IBM

General Information Manual

The IBM 305 RAMAC®

General Purpose Process Control Panel

THE IBM 305 RAMAC

GENERAL PURPOSE PROCESS CONTROL PANEL

© 1958 by International Business Machines Corporation

INDEX

Introduction 1

Constant Storage 3

Unconditional Program Transfers 3

Conditional Program Transfers 4

Selector Setting - Selector 1. 5

Selector Setting - Selector 5. 8

Selector Setting - Selector 6. 9

Selector Setting - Selectors 1, 5 and 6 9

Printing and Punching. 10

Access Arm Control 12

Typing 13

Loading and Starting 13

Card Reading 14

Inquiry 14

Stopping 14

Alteration Switches. 14

Miscellaneous 15

Test Deck. 15

Comparison of Programs 15

Illustrative Program 16

Program Exit Timings 18

Program Exits and Other Elements Not Used 19

Wiring Diagrams. 21

Functional Diagrams 38

INTRODUCTION

This manual presents a general approach to 305 RAMAC programming which uses the General Purpose Process Control Panel. This general approach eliminates Process Control Panel wiring for individual programs and thereby reduces the over-all time for program preparation.

Two of the major factors in the cost of placing an application on the 305 RAMAC are the programming time required to write the program and the machine time required to debug the program. For the 305 RAMAC a technique of programming has been developed to reduce these factors. This technique centers about the use of a single, permanently wired, Process Control Panel called the General Purpose Process Control Panel or General Purpose Board. Each Program Exit on the General Purpose Board performs one or more functions or tests and always the same functions or tests on all applications with which the General Purpose Board is used. For example, Program Exit F causes a card to feed and a program advance.

A program written for use with the General Purpose Board will usually require more execution time than one written for use with a panel wired specifically for the program. An example of this is the Sample Program, Payroll Application, form no. 22-1200-1. When this program was rewritten for use with the General Purpose Board, execution time increased by about 40%. A comparison of this program using the General Purpose Board and using a board specifically wired for the program may be found on page 15. In general, the additional execution time required will vary between 30% and 50%.

The General Purpose Board should be considered for programs in which program execution time is not a primary consideration. For example:

1. Programs that are run infrequently; i.e., weekly, monthly, quarterly, etc., or only once. Special reports and unusual revision of file data frequently fall in this class.

2. Programs that are run frequently but whose execution time is very short. Many file maintenance programs are short enough so that the use of the General Purpose Board will take only a few seconds a day additional machine time.

3. Complex programs that exhaust all program exits or some other portion of the Process Control Panel. This difficulty may usually be resolved by generalizing all or part of the

program exit definitions. The General Purpose Board is an immediately available development of this "generalizing" philosophy.

4. Programs in which the input, output and/or seek time is in excess of the time required to execute the instructions on the 305 Processing Drum. Utility programs frequently fall within this classification. The General Purpose Board will normally take no additional time in these programs.

The General Purpose Process Control Panel is designed to work on any 305 RAMAC; that is, no optional features are required.

On the General Purpose Board each Program Exit is uniquely defined. Programming time is reduced because of consistency. When a programmer wishes to initiate a particular function or test, he need only use the appropriate coding. In addition to the time saved during the writing and documentation of the program, process control panel wiring, debugging, and associated expenses are completely eliminated. For example, the following instruction, when used with the General Purpose Board, will always cause selector 1 to be picked up if accumulator 2 is minus.

From	To	No. of Char.	P	Q
Z2M	-99	01	J	

On the following pages the various Program Exit definitions are given and other General Purpose Board wiring is explained. The programmer must, of course, work within the framework of the Program Exit definitions. These definitions have been chosen so that they will be easy to remember and to exploit the logical facilities of the 305 RAMAC.

CONSTANT STORAGE

When using the General Purpose Process Control Panel, one track of the process drum must be used to store constants. This track must not be disturbed during the running of the program and usually will be loaded when the program itself is loaded. The layout of this track is as follows:

.ABCDEFGHI	$JKLMNOPQR	,/STUVWXYZ
00	10	20 29

#1234567890	bbb&¤-*%@	bAbBbbEbF	bbJKbbObQ
30	40	50	60 69

bbbbOBbHbY	bbbb2bb5b8	bbbb-bb%bb
70	80	90 99

b = blank character

Since each RAMAC character is represented at least once, this track may be used as a source for constants needed by a program.

In the descriptions which follow, it is assumed that the constants have been stored on track Z.

UNCONDITIONAL PROGRAM TRANSFERS

Program Exits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9

An instruction of the form

From	To	No. of Char.	P	Q
Z(h+2)t	-99	01	u	

causes a program transfer to step number htu, where h, t, and u represent the hundreds, tens and units digits of the step respectively.

Examples:	From	To	No. of Char.	P	Q
To transfer to step 187	Z38	-99	01	7	
To transfer to step 038	Z23	-99	01	8	

Program Exits Z, D

Program Exit Z transfers the program to step 190; Program Exit D transfers the program to step 199.

CONDITIONAL PROGRAM TRANSFERS

Program Exits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9

An instruction of the form

From	To	No. of Char.	P	Q
Zht	-99	01	u	

causes a program transfer to step number htu, where h, t, and u represent the hundreds, tens and units digits respectively, if selector 1 is transferred. If selector 1 is normal, a program advance occurs. Selector 1 is set as described under "Selector Setting," pages 5 & 9.

Program Exit @

An instruction of the form

From	To	No. of Char.	P	Q
Z(h+2)t	-99	01	@	

causes a program transfer to step number htu, where h, t, and u represent the hundreds, tens, and units digits of the step, respectively. u is the number of the first accumulator in ascending order (0, 1, 2, . . ., 9) which indicates either plus or zero. If all accumulators indicate minus, a program advance occurs.

Examples:	From	To	No. of Char.	P	Q
To transfer to step 14u:	Z34	-99	01	@	

If accumulators 0, 1 and 2 are minus and 3 is plus, the transfer will be to step 143. If all the accumulators were plus, the transfer will be to step 140.

Program Exit ⌑

An instruction of the form

From	To	No. of Char.	P	Q
Z(h+2)t	-99	01	⌑	

causes a program transfer similar to that caused by Program Exit @. In this case the field selectors are tested, in ascending order, for an equal condition. If all the field selectors indicate unequal, a program advance occurs.

Program Exit C

Program Exit C causes a program transfer to step 199 if accumulator 9 is minus. If accumulator 9 is not minus, a program advance occurs.

SELECTOR SETTING - SELECTOR 1

Program Exits E, U

These exits cause selector 1 to be picked up or dropped out depending on the status of the compare selector.

Program Exit E: Pick up selector 1 if compare selector indicates Equal
Drop out selector 1 if compare selector indicates unequal

Program Exit U: Pick up selector 1 if compare selector indicates Unequal
Drop out selector 1 if compare selector indicates equal

Example:	From	To	No. of Char.	P	Q
To compare positions 00-04 on track W with	W04	K09	05	E	1
positions 05-09 on track K and transfer to step 048 if equal and program advance if unequal.	Z04	-99	01	8	

Program Exits B, N

These exits cause selector 1 to be picked up or dropped out depending on the status of the blank transmission selector.

Program Exit B: Pick up selector 1 if transmission is Blank
Drop out selector 1 if transmission is not blank

Program Exit N: Pick up selector 1 if transmission is Not blank
Drop out selector 1 if transmission is blank

Example:	From	To	No. of Char.	P	Q
To test positions 0-9 of track W for blanks. If	W09	K09	10	B	
blank, transfer to step 183. If not blank, program advance.	Z18	-99	01	3	

Program Exits V, W

These exits cause selector 1 to be picked up or dropped out depending on the status of the accumulator overflow selector. In addition, both of these exits drop out the accumulator overflow selector after positioning selector 1.

Program Exit V: Pick up selector 1 if there has been an overflow
Drop out selector 1 if there has been no overflow

Program Exit W: Pick up selector 1 if there has been no overflow
Drop out selector 1 if there has been an overflow

Example:	From	To	No. of Char.	P	Q
To transfer to step 069 if there has been no	G10	L06	03	W	
overflow, or to transfer to step 188 if there has	Z06	-99	01	9	
been an overflow.	Z38	-99	01	8	

Program Exit J

An instruction of the form

From	To	No. of Char.	P	Q
Zxy	-99	01	J	

causes selector 1 to be picked up if the statement indicated by xy below is true. If the statement indicated by xy is untrue, selector 1 is dropped out.

xy	Statement
0P	Accumulator 0 is plus.
0M	Accumulator 0 is minus.
0Z	Accumulator 0 is zero.
1P	Accumulator 1 is plus.
1M	Accumulator 1 is minus.
1Z	Accumulator 1 is zero.
.	.
.	.
.	.
9P	Accumulator 9 is plus.
9M	Accumulator 9 is minus.
9Z	Accumulator 9 is zero.
S1	Selector 1 is transferred.
S5	Selector 5 is transferred.
S6	Selector 6 is transferred.
CF	Compare selector indicates equal.
FL	Accumulator overflow selector indicates yes.
LA	Last card selector indicates yes.
A5	Alteration Switch 5 is ON.
BS	Selectors 5 and 6 are both transferred.
NS	Neither Selector 5 nor 6 is transferred.
UN	Position Selector 1 unconditionally.*

*This statement is an exception to the other xy statements in that it is always assumed to be true.

Examples:	From	To	No. of Char.	P	Q
To transfer to step 067 if accumulator 4 is zero,	Z4Z	-99	01	J	
or, if not, to program advance.	Z06	-99	01	7	
To transfer to step 100 if selector 5 is transferred,	ZS5	-99	01	J	
or, if not, to program advance.	Z10	-99	01	0	
To pick up selector 1.	ZUN	-99	01	J	

Program Exit K

This Program Exit has the opposite meaning of Program Exit J; that is, selector 1 is dropped out if the statement indicated by xy is true and selector 1 is picked up if the statement indicated by xy is untrue.

Examples:	From	To	No. of Char.	P	Q
To reverse the position of selector 1.	ZS1	-99	01	K	
To drop out selector 1.	ZUN	-99	01	K	
To transfer to step 134 if accumulator 8 is minus	Z8P	-99	01	K	
or zero, or, if positive to program advance.	Z13	-99	01	4	

SELECTOR SETTING - SELECTOR 5

Program Exits L, M

Program Exits L and M are identical with exits J and K respectively with the exception that selector 5 is picked up or dropped out rather than selector 1. The S5, BS and NS statements when used with Exits L and M cause selector 5 to be tested and then positioned.

SELECTOR SETTING - SELECTOR 6

Program Exits P, Q

Program Exits P and Q are identical with exits J and K respectively with the exception that selector 6 is picked up or dropped out rather than selector 1. The S6, BS and NS statements when used with Exits P and Q cause selector 6 to be tested and then positioned.

SELECTOR SETTING - SELECTORS 1, 5 and 6

Program Exit A

This Program Exit causes the zone of the character in the character selector to be stored in selectors 1, 5, and 6 as follows:

	Selector 1	Selector 5	Selector 6
12 Zone	Transferred	Transferred	Transferred
11 Zone	Transferred	Transferred	Normal
0 Zone	Transferred	Normal	Transferred
No Zone	Normal	Transferred	Transferred

As a memory aid, note that for 12, 11 and 0 Zones, selectors 5 and 6 are transferred or normal according to whether the X and 0 bits are present or absent in the 305 RAMAC character code and that selector 1 is transferred if the character has zoning.

Program Exit A is designed as an aid to character analysis and may easily be used, for example, to cause branching to any one of several different subroutines based on the analysis of a particular character.

Example:	Step No.	From	To	No. of Char.	P	Q
To analyze the character in position 30 of track W and transfer to step 148 if it has a 12 zone, to step 114 if it has an 11 zone, to step 183 if it has a	033	W30	-99	01	A	
	034	Z04	-99	01	2	
	.					
	. Numerical character processing routine					
	.					
	041	Z24	-99	01	7	

Example:	Step No.	From	To	No. of Char.	P	Q
0 zone, and to step	042	ZBS	-99	01	J	
137 if it has no	043	Z14	-99	01	8	
zone.	044	ZS5	-99	01	J	
	045	Z11	-99	01	4	
	046	Z38	-99	01	3	
	047	To continue program				

Concerning the "Selector Setting" exists in general, selector 1 is the only selector that may be used to control a conditional program transfer. Thus, in practice, even though decisions may be "stored" in selectors 5 and 6, normally these decisions will be moved to selector 1 in order to provide for a conditional program transfer.

PRINTING AND PUNCHING

Program Exits -, $, *, #, and & are used to control printing and punching. Selector 4 and cycle delay 6 are used in connection with these exits as controlling devices for the printer. In addition, certain of the print and punch communication hubs are connected to these exits so that with appropriate wiring on the punch control panel, certain of these exits may be joined. A description of the Program Exits and a general discussion of how these exits might be used follows.

Program Exit -

This exit tests selector 4 and causes the following:

If selector 4 is normal:

1. printing.
2. selector 1 to be dropped out.
3. a program advance.
4. print communication 4 to be impulsed through a distributor.
5. punch communication 4 to be impulsed directly.

If selector 4 is transferred:

1. selector 1 to be picked up.
2. a program advance.
3. print communication 5 to be impulsed through a distributor.

Other wiring concerning the 310 printer:

Print communication 7 is wired to the pickup of selector 4.

Program Exit - is designed primarily as an aid in the printing of page totals on a report. For example, assume the overflow (OF) exit on the printer is wired to print communication 7. Thus, if no overflow has occurred, printing will take place, and if an overflow has occurred, no printing will take place and selector 1 is picked up to provide for a transfer to a total routine. Communication with the printer and punch panels is provided.

Program Exit $

This exit causes:

1. printing.
2. selector 4 to be dropped out.
3. a program advance.
4. print communication 8 to be impulsed through a distributor.
5. punch communication 8 to be impulsed directly.

This exit is designed primarily for the printing of headings or page totals. (See Program Exit - above.) Communication with the printer and punch panels is provided.

Program Exit *

This exit is wired to:

1. print communication 1 through a distributor.
2. cycle delay 6 in.

Cycle delay 6 out causes:

1. printing.
2. selector 4 to be dropped out.
3. a program advance.
4. print communication 2 to be impulsed through a distributor.
5. punch communication 1 to be impulsed directly.

Program Exit * is designed primarily to provide for skipping before printing. Communication with the printer and punch panels is provided.

Program Exit &

This exit causes:

1. punching.
2. a program advance.
3. punch communication 7 to be impulsed through a distributor.
4. punch communication 3 to be impulsed directly.

Program Exit #

This exit causes:

1. punching.
2. a program advance.
3. punch communication 6 to be impulsed through a distributor.
4. punch communication 2 to be impulsed directly.

In the descriptions of the exits used to cause printing and punching, it may have been noted that direct wiring (not through a distributor) has been provided to a particular communication hub on the punch panel. Thus if a communication hub on the punch control panel is impulsed, the functions of the exit to which it is directly connected will be performed. By joining communication hubs on the punch panel it is possible to have one Program Exit perform the function of another Program Exit as well as its own function. Careful study is necessary in order to take full advantage of this feature. As a convenient reference, the table below shows which Program Exits are wired to which communication hubs on the punch control panel.

Program Exit	Wired Directly to Punch Communication
-	4①
$	8
*	1②
&	3
#	2

① Punch communication 4 is wired to the normal side of selector 4 and the functions are performed as though selector 4 were normal regardless of the status of selector 4.

② The functions performed are those performed by cycle delay 6 out.

ACCESS ARM CONTROL

Program Exit R

This exit causes a record advance and then a program advance.

Program Exit S

This exit causes a skip to record n where n is the number of the first field compare selector in ascending order (0, 1, . . ., 9) which indicates equal. If a skip-to-record occurs, selector 1 is dropped out. If no skip-to-record occurs, that is all the field compare selectors indicate unequal, selector 1 is picked up. A program advance occurs in either case.

Example:	From	To	No. of Char.	P	Q
To skip to the record indicated by the field compare selectors and program advance. If all the field compare selectors indicate unequal, transfer to step 135.	W99 Z13	X99 -99	00 01	S 5	2

TYPING

Program Exit / (See also Program Exit Y)

This exit causes typing if Alteration Switch 4 is OFF. If Alteration Switch 4 is ON, this exit has no effect.

LOADING AND STARTING

Program Exit X

This exit is wired to Copy In.

Program Exit Y

This exit performs the functions of Program Exit X and Program Exit /; e. g., it is wired to Copy In and causes typing if Alteration Switch 4 is OFF.

Copy Out tests the Last Card Selector. If NO, a card is fed and the program transfers to step 190. If YES, a reset stop occurs.
The Start hub resets all selectors, drops out the Accumulator Overflow Selector and transfers the program to step 000.

CARD READING (SEE ALSO "LOADING AND STARTING")

Program Exit F

This exit causes a card to feed.

Program Exit G

This exit tests the Last Card Selector. If NO, a card is fed, and selector 1 is dropped out. If YES, selector 1 is picked up, but a card is not fed.

Example:	From	To	No. of Char.	P	Q
To transfer to a final total routine at step 168 if the last card has been processed. If not, program advance.	K99	W99	00	G	
	Z16	-99	01	8	

INQUIRY

Program Exit I

This exit is wired to Program Advance through the Inquire In-Out hubs. The Inquire ON switch is jackplugged.

STOPPING

Program Exit.

This exit is wired to Stop and to Program Advance.

Program Exit H

This exit tests the compare selector. If equal, a program advance occurs. If unequal, a stop occurs.

ALTERATION SWITCHES

Switch 1	ON: File ON. OFF: File OFF.
Switch 2	Not used.
Switch 3	Not used.

Switch 4	ON: Prevents typing with Program Exits / and Y. OFF: Causes typing with Program Exits / and Y.
Switch 5	Used as an external logical control with Program Exits J, K, L, M, P, and Q.
Switch 6	ON: ALC ON. OFF: ALC OFF.

MISCELLANEOUS

ITI is wired.
Inquire is wired On. (See Program Exit I, page 14).
Start (See "Loading and Starting," page 13).

TEST DECK

A test deck which will determine if the General Purpose Board is correctly wired and instructions for use of the test deck are available, on a basis of one per installation, from:

305 RAMAC Program Librarian
Commercial Systems and Development
IBM Corporation
590 Madison Avenue
New York 22, N. Y.

COMPARISON OF PROGRAMS

When the "Sample Program Payroll Application" (Form No. 22-1200-1) was programmed for use with the General Purpose Board, the following changes were tabulated:

	Using a specifically wired board	Using the General Purpose Board	Ratio
Number of instructions	97	108	1.11
Average processing time per card	3.42	4.83	1.41

ILLUSTRATIVE PROGRAM

The following short program illustrates the use of the General Purpose Board in a typical file-searching routine.

Salesman records (disk addresses 12500 to 12999) are examined, and a card is punched for each salesman whose year-to-date sales equal or exceed quota.

Prog. Step	FROM		TO		No. Char-acters	Control	Description	Skip To Prog. Step	Time in milli-seconds
	Track	Position	Track	Position					
							Initialization		
135	E 1	4	L 9	9	0 5	5	Reset add first disk address in accumulator 9.		
136	L 9	9	J 9	9	0 5		Seek first salesman record		
137	Z 2	4	C 8	8	0 1		Move a "U" to P-flag position of step 138.		
							Program Loop		
138	L 9	9	E 1	9	0 5	U 1	Compare this address to final address; if unequal, transfer to step	141	
139	Z 1	4	- 9	9	0 1	1			
140	Z 0	0	C 8	8	0 1		If this is the final address, move a period (stop) to P-flag position of step 138.		
141	R 9	9	W 9	9	0 0		Move salesman record to working track.		
142	L 9	9	W 0	4	0 5	H 1	Compare accumulator 9 with record address; if unequal, stop.		
143	Z 3	1	L 9	9	0 1		Add 1 to accumulator 9.		
144	L 9	9	J 9	9	0 5		Seek next salesman record.		
145	W 5	4	L 8	9	0 5	5	Reset add QUOTA in accumulator 8.		

Prog. Step	FROM			TO			No. Char-acters		Control		Description	Skip To Prog. Step	Time in milli-seconds
	Track	Position		Track	Position								
146	W	5	9	M	8	9	0	5			Subtract YEAR -TO-DATE		
											SALES from accumulator 8.		
147	Z	8	P	-	9	9	0	1	J		If QUOTA exceeds YEAR-TO-		
148	Z	1	3	-	9	9	0	1	8		DATE. SALES (accumulator 8		
											plus) transfer to step	138	
149	W	9	9	S	9	9	0	0	#		Punch record of salesman who		
											has equaled or exceeded		
											quota.		
150	Z	3	3	-	9	9	0	1	8		Transfer to step	138	
151	1	2	5	0	0	1	2	9	9	9	Constants (first and final disk		
											addresses)		

PROGRAM EXIT TIMINGS

The following timing chart showing the execution time of each Program Exit, as used in this manual, is presented here to be used as a guide in timing a program written for use with the General Purpose Board. The times shown are in addition to the normal 30 m/s required to execute an instruction.

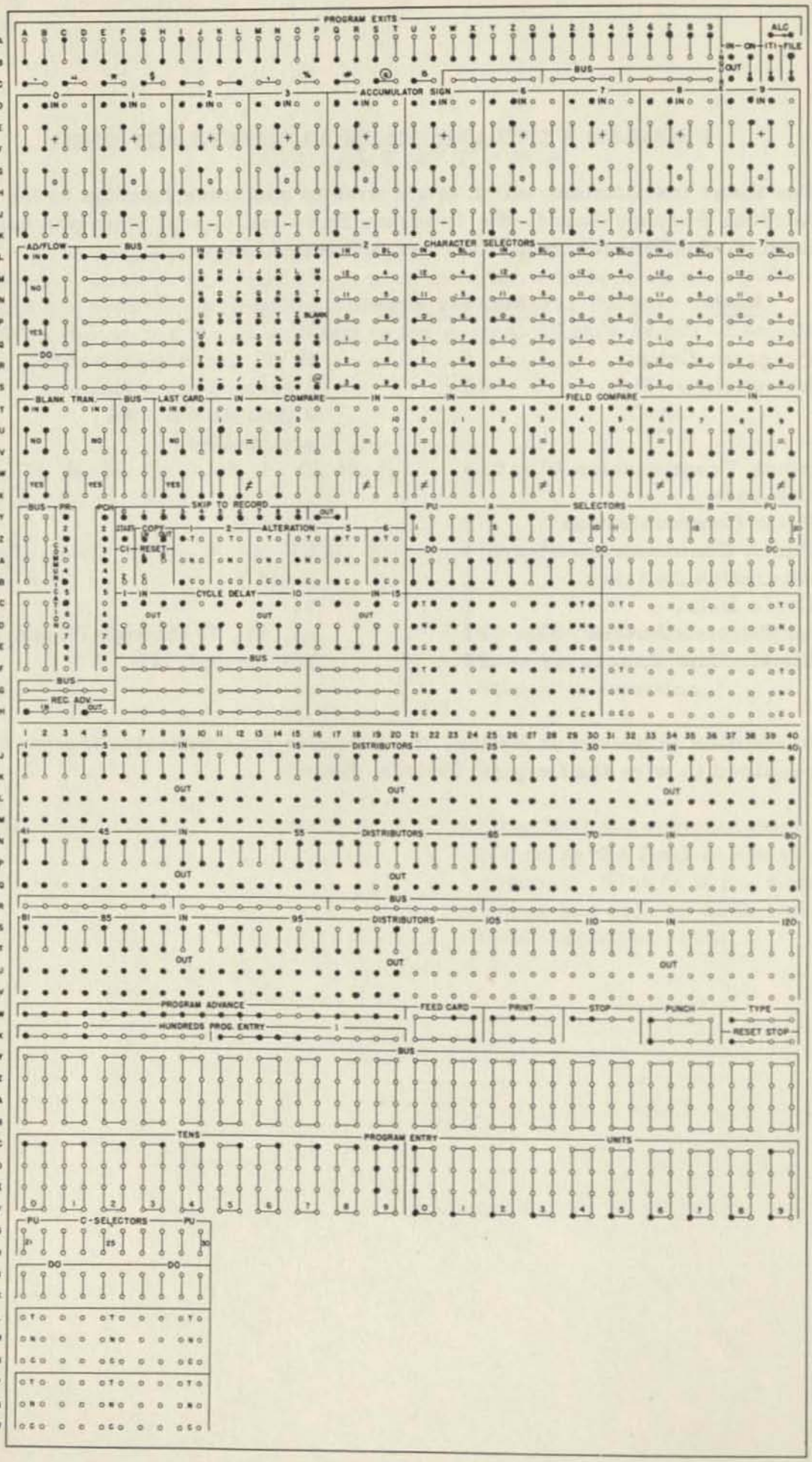
Program Exits	Time
E, U, B, N, C, D, Z, F, G, H, I, ., &, #, —, $, 1	20 m/s
0-9, J, K, L, M, P, Q, V, W, S, R, A	50 m/s
X, Y	70 m/s (This time includes the time required to execute the "Copy" instruction.)
*	20 m/s normally (see functional diagrams)
@, ¤	20 m/s if the accumulators are minus or the field compares unequal. 50 m/s if the accumulators are plus or zero and the field compares equal.

PROGRAM EXITS AND OTHER ELEMENTS NOT USED

For those installations which might wish to add some function to the General Purpose Board, the Program Exits and logical elements not used by the General Purpose Board are listed below.

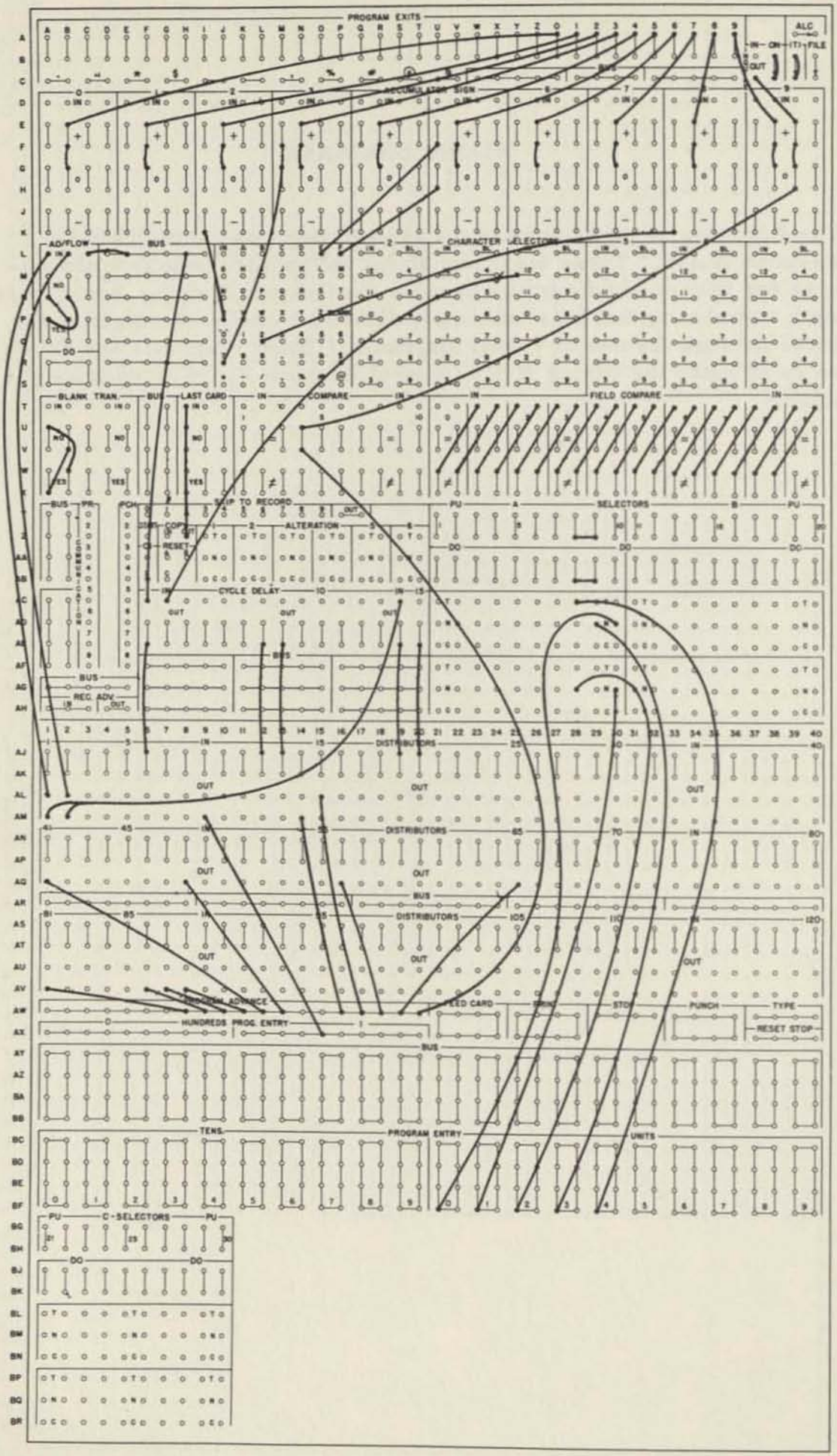
Item	Number Free	Identity
Program Exits	4	O, T, ,(comma), and %
Accumulator Sign	19	2 each on accumulators 0 through 8. 1 on accumulator 9.
Blank Transmission	3	
Compare	6	
Print Communication	2	3 and 6
Punch Communication	1	5
Alteration Switches	2	2 and 3
Cycle Delay	1	11
Single Distributors	11	43, 59, 70 through 77 and 79.
Double Distributors	20	101 through 120.

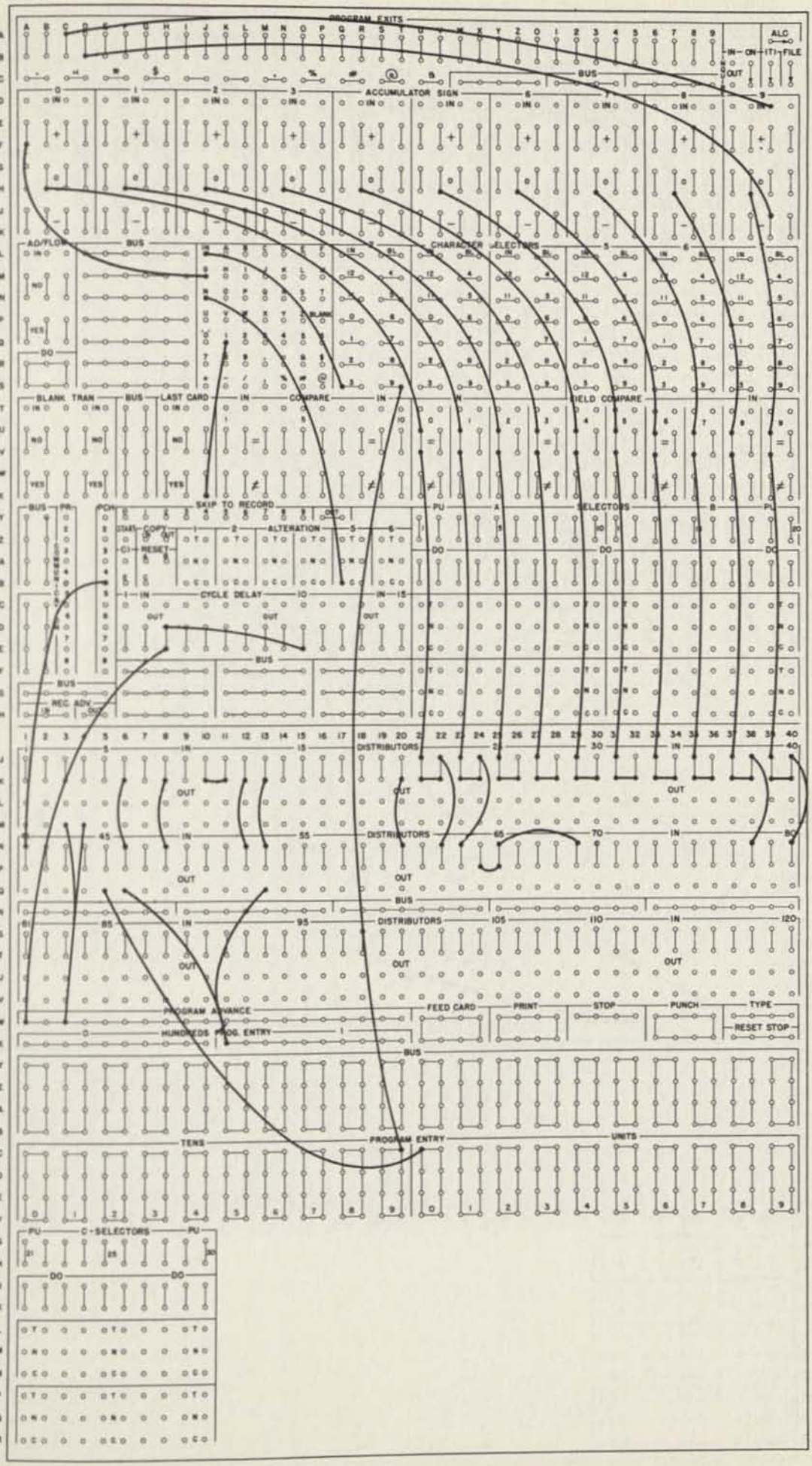
In addition to the above listing, the following page indicates the hubs required to wire the General Purpose Board.

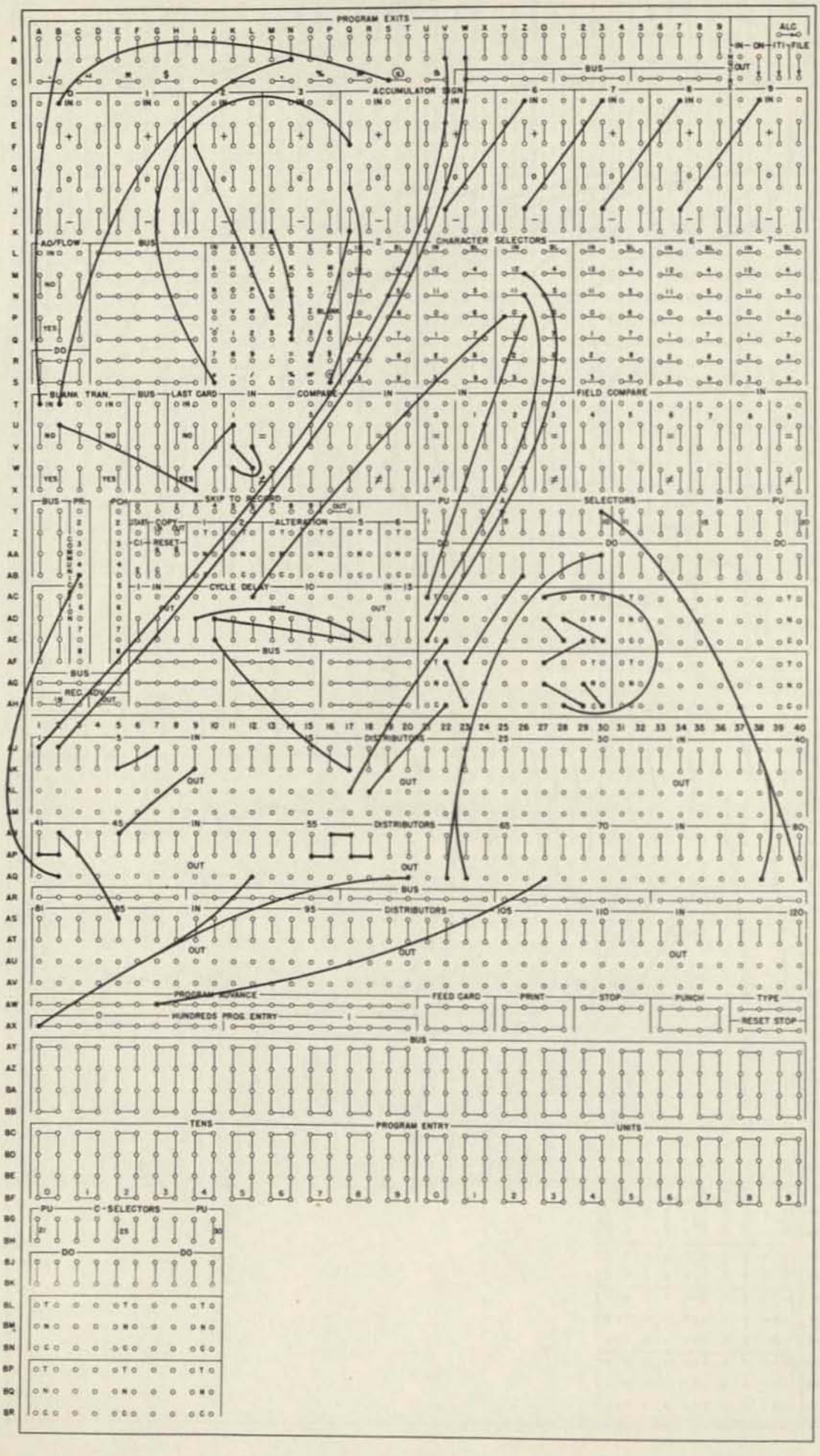

PROGRAM EXITS
ALC
ACCUMULATOR SIGN
CHARACTER SELECTORS
BUS
AD/FLOW
BLANK TRAN.
LAST CARD
COMPARE
FIELD COMPARE
SKIP TO RECORD
CYCLE DELAY
REC. ADV.
DISTRIBUTORS
PROGRAM ADVANCE
FEED CARD
PRINT
STOP
PUNCH
TYPE
RESET STOP
HUNDREDS PROG. ENTRY
TENS
PROGRAM ENTRY
UNITS
C-SELECTORS

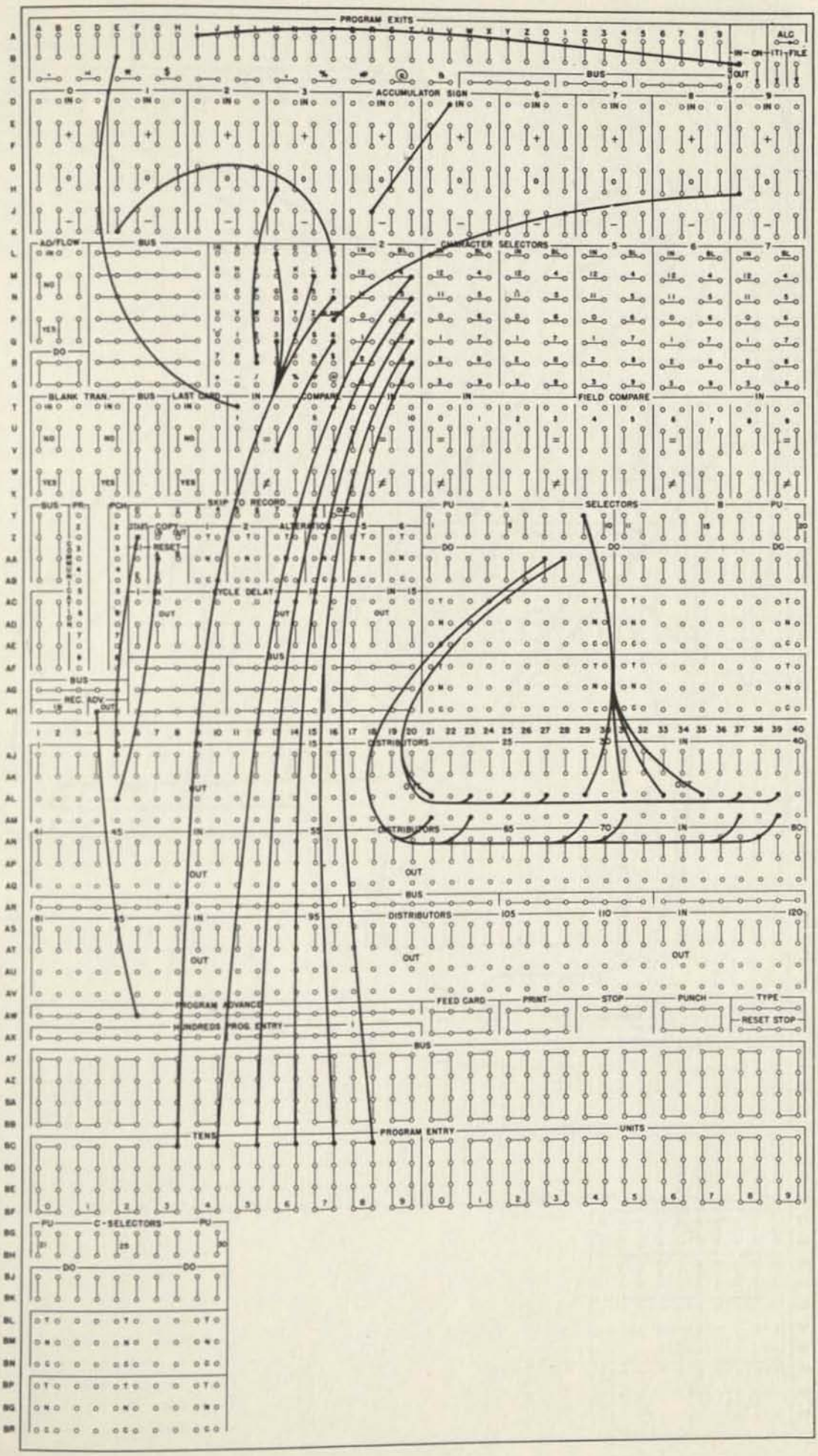
WIRING DIAGRAMS

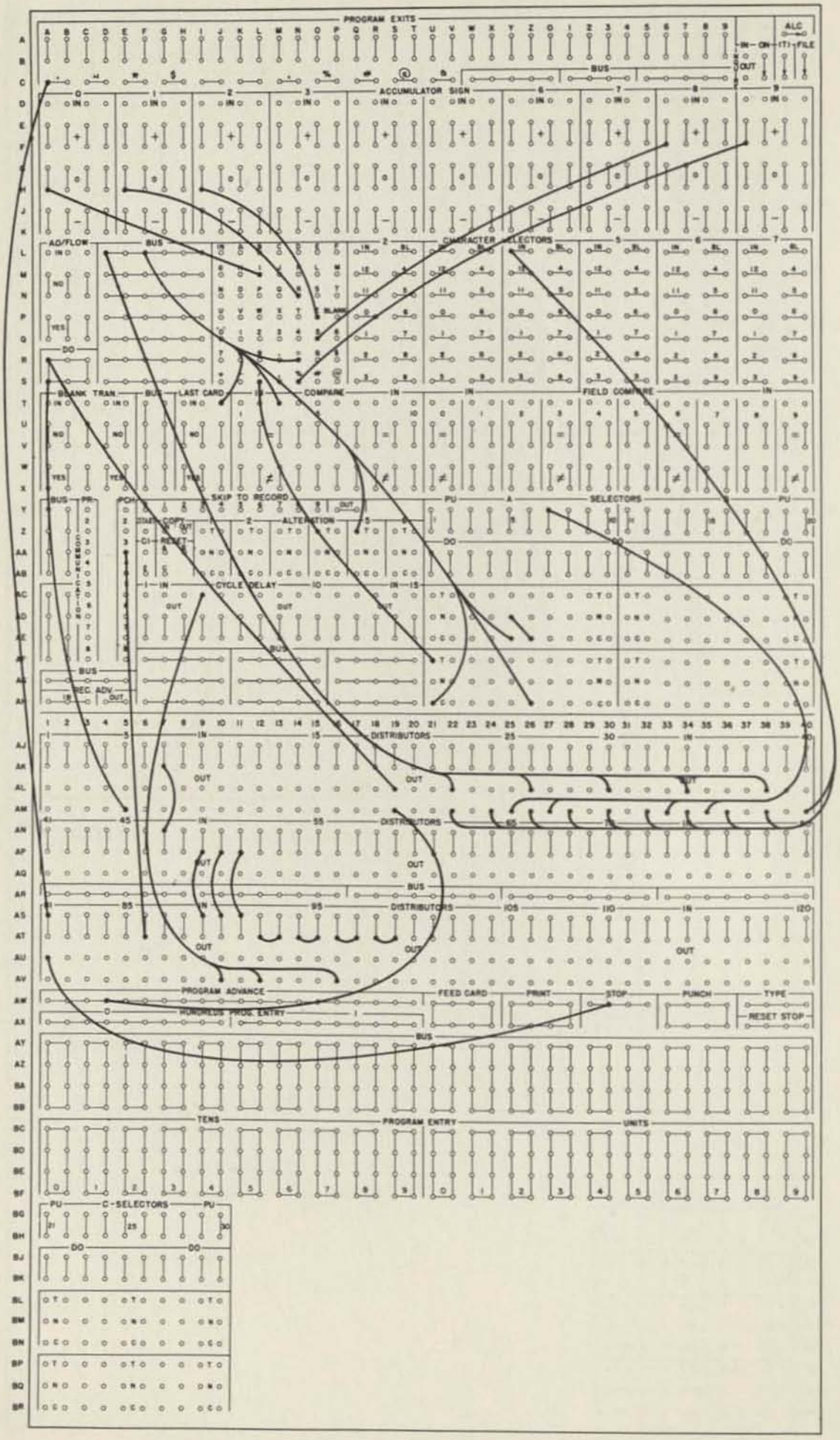
The following 16 pages contain the wiring diagrams for the General Purpose Board. These diagrams do not follow recommended procedures for presenting diagrams because the General Purpose Board is of such complexity that it is extremely difficult to follow the logic of the panel through the use of wiring diagrams. These diagrams are drawn so as to make the wiring of the General Purpose Board as easy as possible.

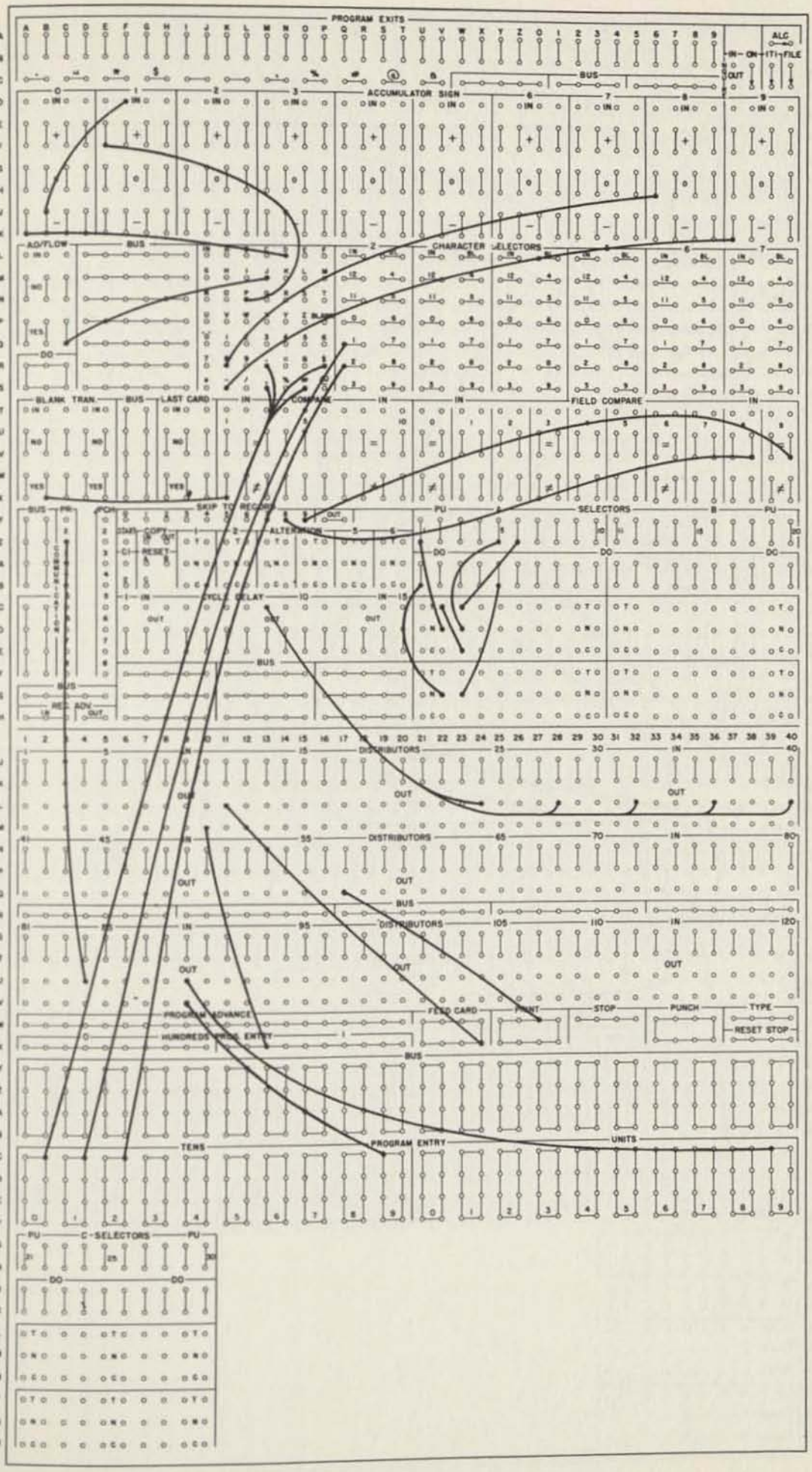

PROGRAM EXITS
ALC
IN
ON
OUT
FILE
ACCUMULATOR SIGN
CHARACTER SELECTORS
AD/FLOW
BUS
DO
BLANK TRAN.
LAST CARD
COMPARE
FIELD COMPARE
NO
YES
SKIP TO RECORD
PCH
PR
CONT. COPY
RESET
ALTERATION
CYCLE DELAY
REC. ADV.
PU
A
SELECTORS
B
DISTRIBUTORS
PROGRAM ADVANCE
FEED CARD
STOP
PUNCH
TYPE
RESET STOP
HUNDREDS PROG. ENTRY
TENS
PROGRAM ENTRY
UNITS
C-SELECTORS

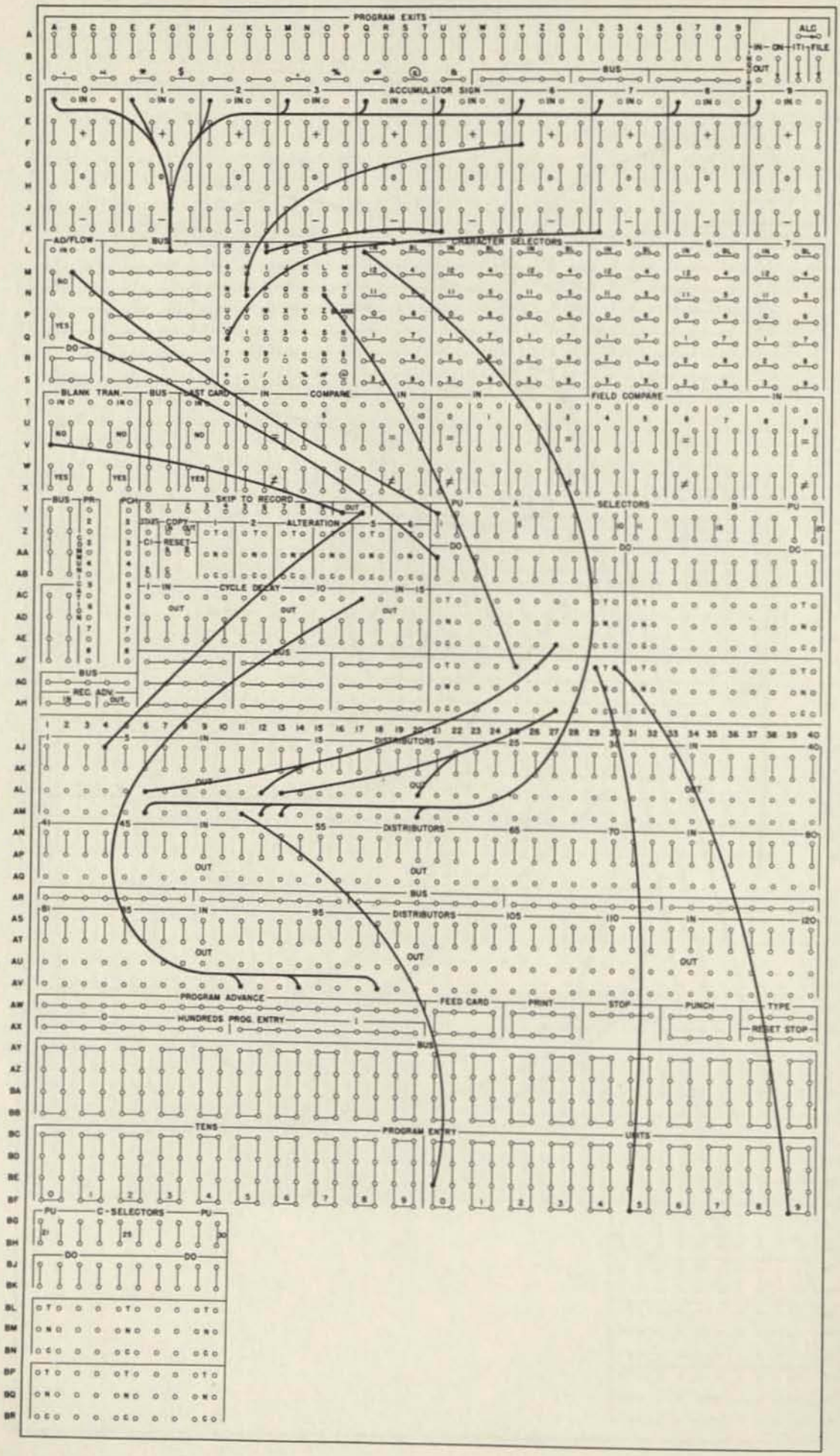

PROGRAM EXITS
ALC
IN
ON
IT
FILE
BUS
OUT
ACCUMULATOR SIGN
AD/FLOW
CHARACTER SELECTORS
BLANK TRAN
LAST CARD
COMPARE
FIELD COMPARE
SKIP TO RECORD
COPY
ALTERATION
RESET
CYCLE DELAY
SELECTORS
REC ADV
DISTRIBUTORS
PROGRAM ADVANCE
HUNDREDS PROG. ENTRY
FEED CARD
PRINT
STOP
PUNCH
TYPE
RESET STOP
PROGRAM ENTRY
TENS
UNITS
C-SELECTORS

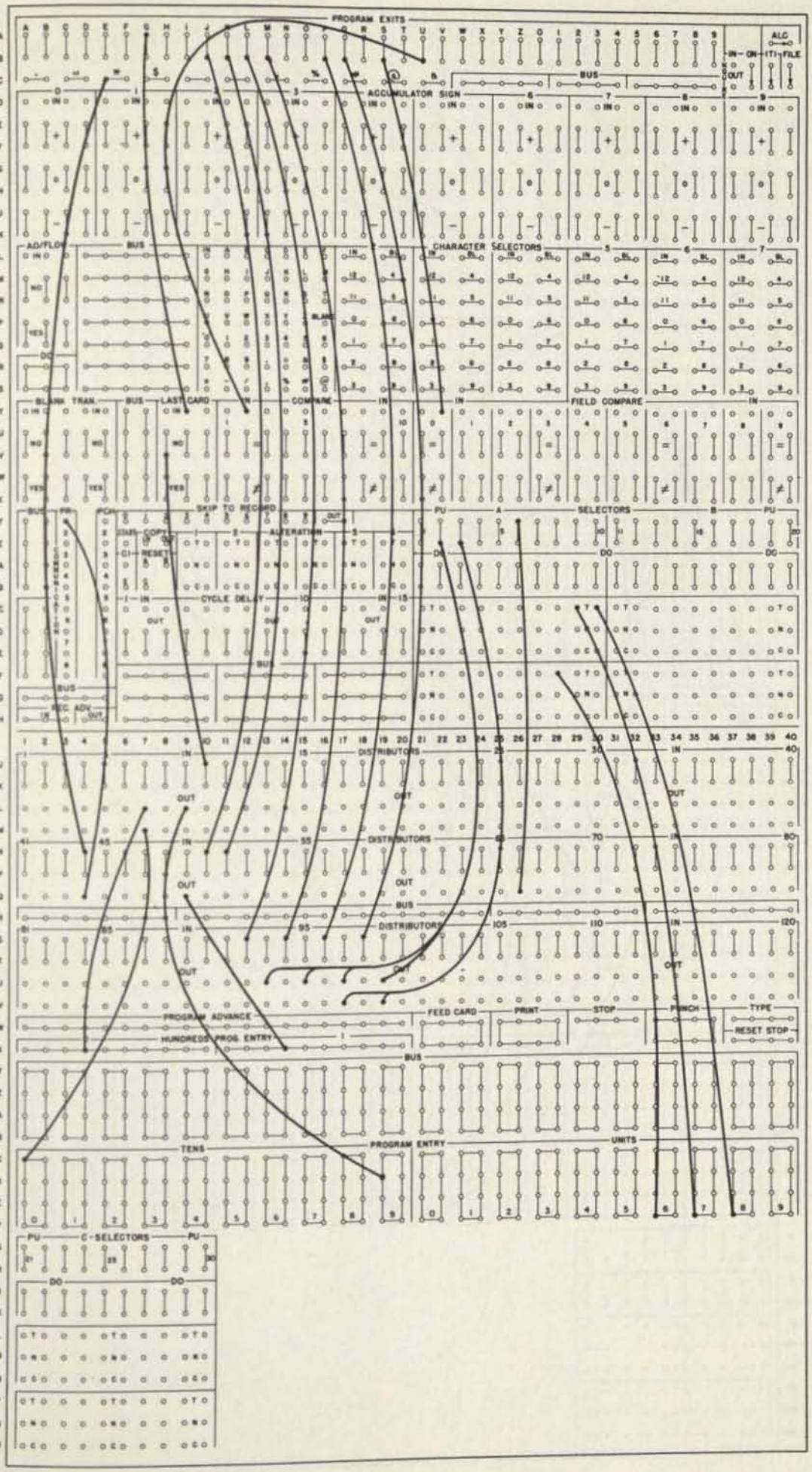

PROGRAM EXITS
ALC
IN
ON
ITI
FILE
OUT
BUS
ACCUMULATOR SIGN
AD/FLOW
CHARACTER SELECTORS
BLANK TRAN.
LAST CARD
COMPARE
FIELD COMPARE
SKIP TO RECORD
ALTERATION
CYCLE DELAY
SELECTORS
DISTRIBUTORS
PROGRAM ADVANCE
FEED CARD
PRINT
STOP
PUNCH
TYPE
RESET STOP
HUNDREDS PROG. ENTRY
TENS
PROGRAM ENTRY
UNITS
C-SELECTORS
PU
DO
YES
NO

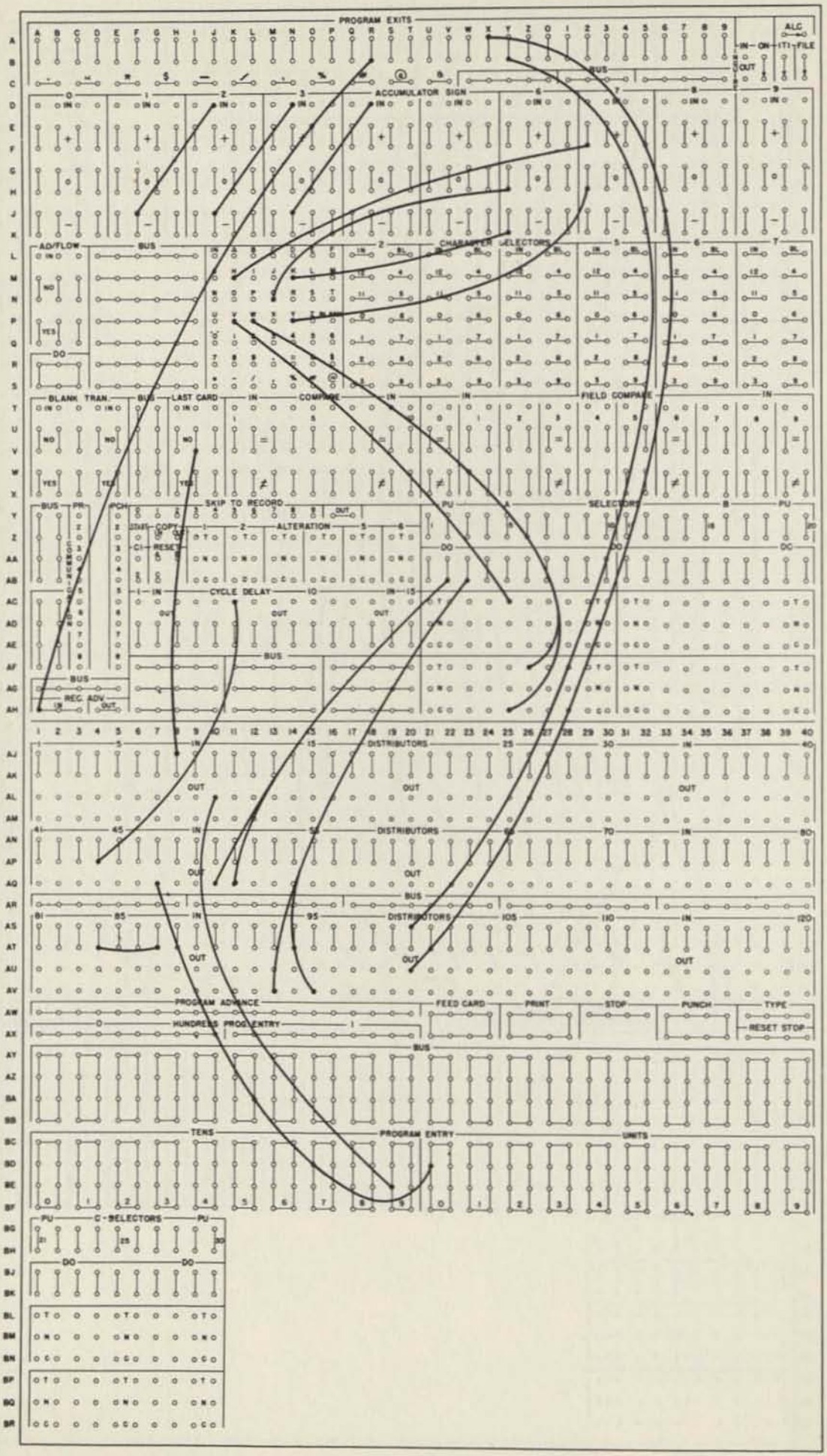

PROGRAM EXITS
ALC
IN
ON
ITI
FILE
OUT
BUS
ACCUMULATOR SIGN
AD/FLOW
CHARACTER SELECTORS
DO
BLANK TRAN.
LAST CARD
COMPARE
FIELD COMPARE
YES
NO
SKIP TO RECORD
SELECTORS
PU
A
B
CYCLE DELAY
ALTERATION
REC. ADV.
DISTRIBUTORS
PROGRAM ADVANCE
HUNDREDS PROG. ENTRY
FEED CARD
PRINT
STOP
PUNCH
TYPE
RESET STOP
TENS
PROGRAM ENTRY
UNITS
C-SELECTORS

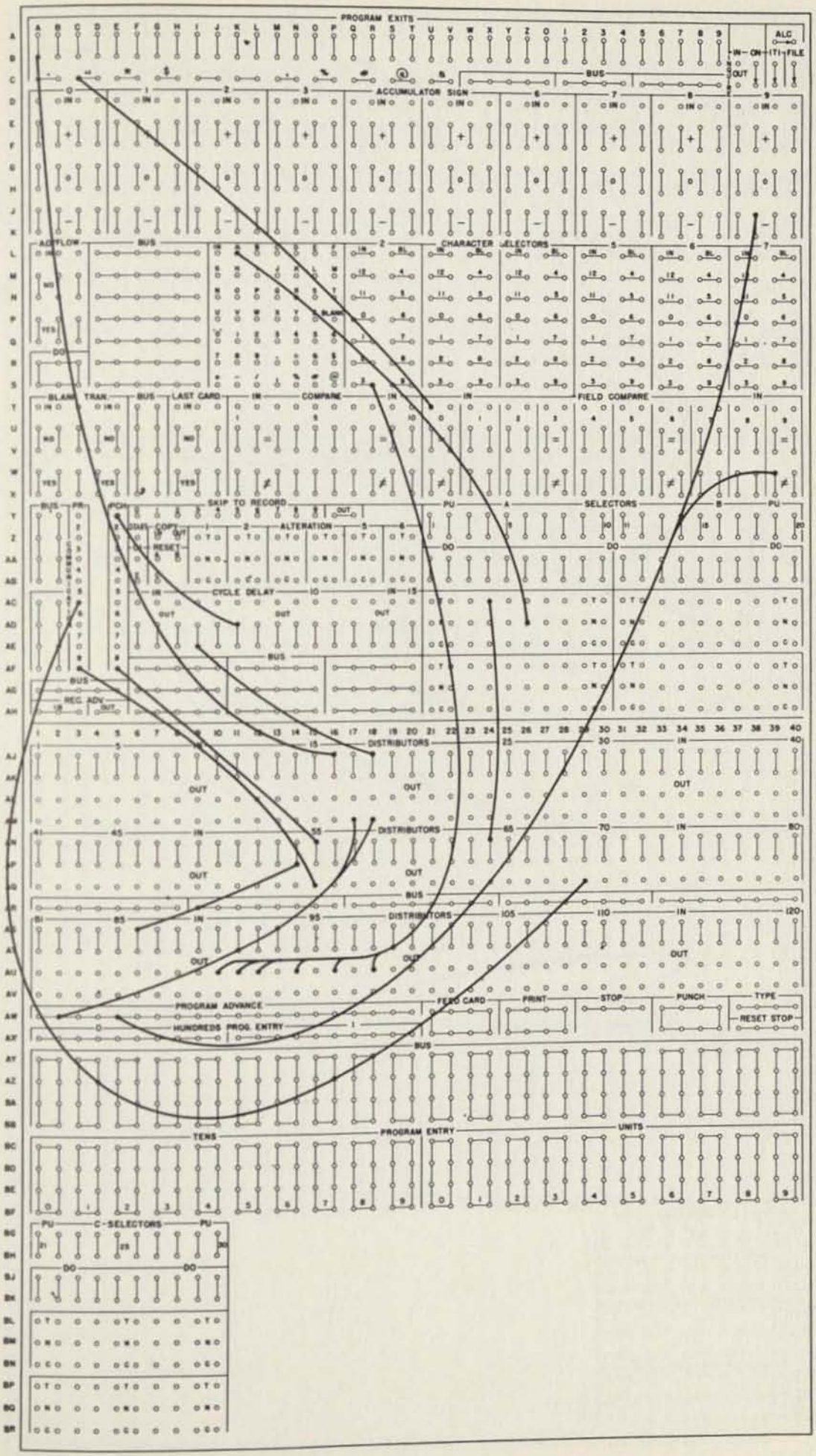

PROGRAM EXITS
ALC
ACCUMULATOR SIGN
CHARACTER SELECTORS
AD/FLOW
BUS
BLANK TRAN
LAST CARD
COMPARE
FIELD COMPARE
SKIP TO RECORD
SELECTORS
CYCLE DELAY
REC. ADV.
DISTRIBUTORS
PROGRAM ADVANCE
FEED CARD
PRINT
STOP
PUNCH
TYPE
HUNDREDS PROG. ENTRY
RESET STOP
TENS
PROGRAM ENTRY
UNITS
C-SELECTORS

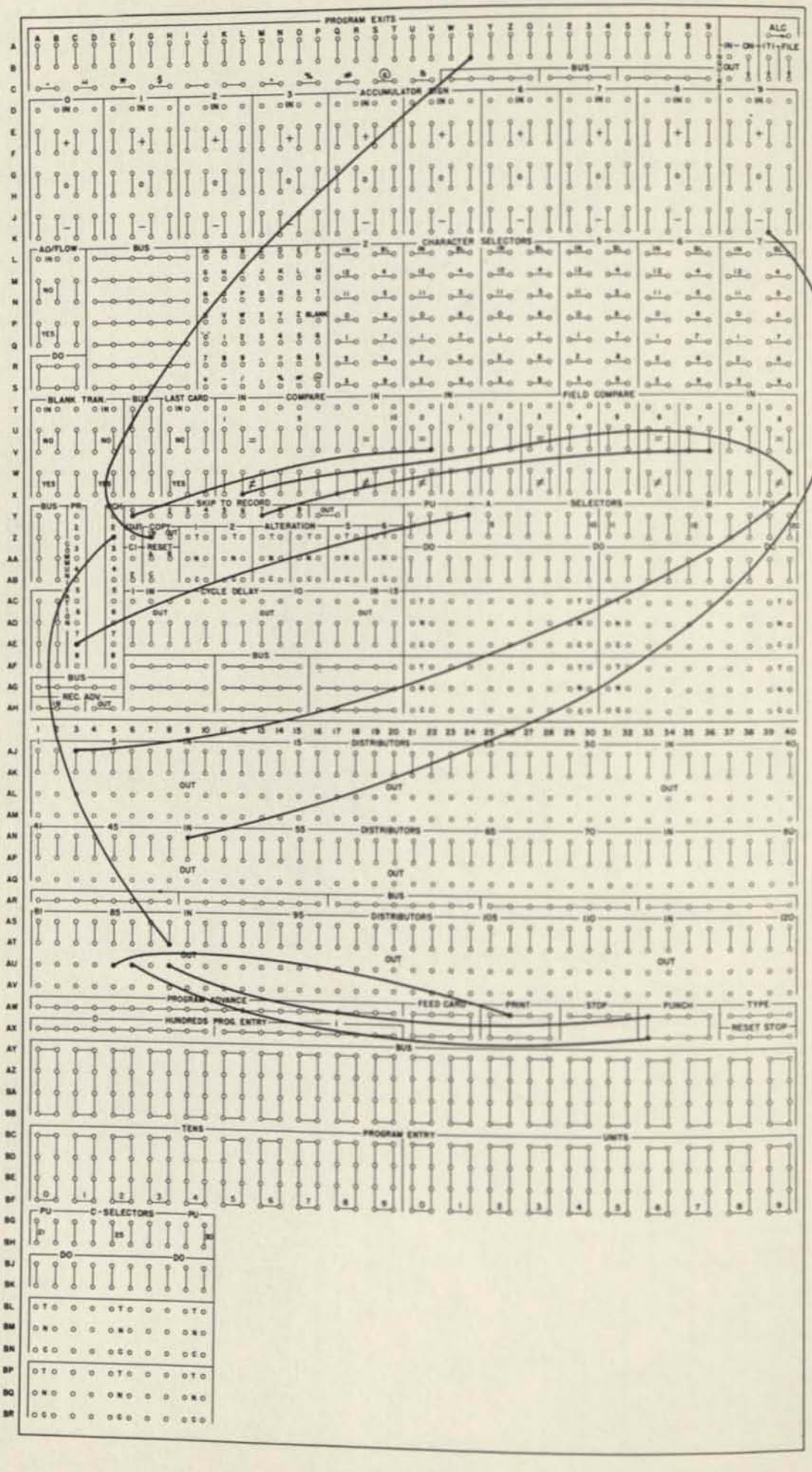

PROGRAM EXITS
ACCUMULATOR SIGN
BUS
ALC
IN
OUT
FILE
ADJ/FLOW
CHARACTER SELECTORS
BLANK TRAN.
LAST CARD
COMPARE
FIELD COMPARE
SKIP TO RECORD
ALTERNATION
CYCLE DELAY
SELECTORS
DISTRIBUTORS
PROGRAM ADVANCE
FEED CARD
PRINT
STOP
PUNCH
TYPE
RESET STOP
HUNDREDS
TENS
UNITS
PROGRAM ENTRY
C-SELECTORS

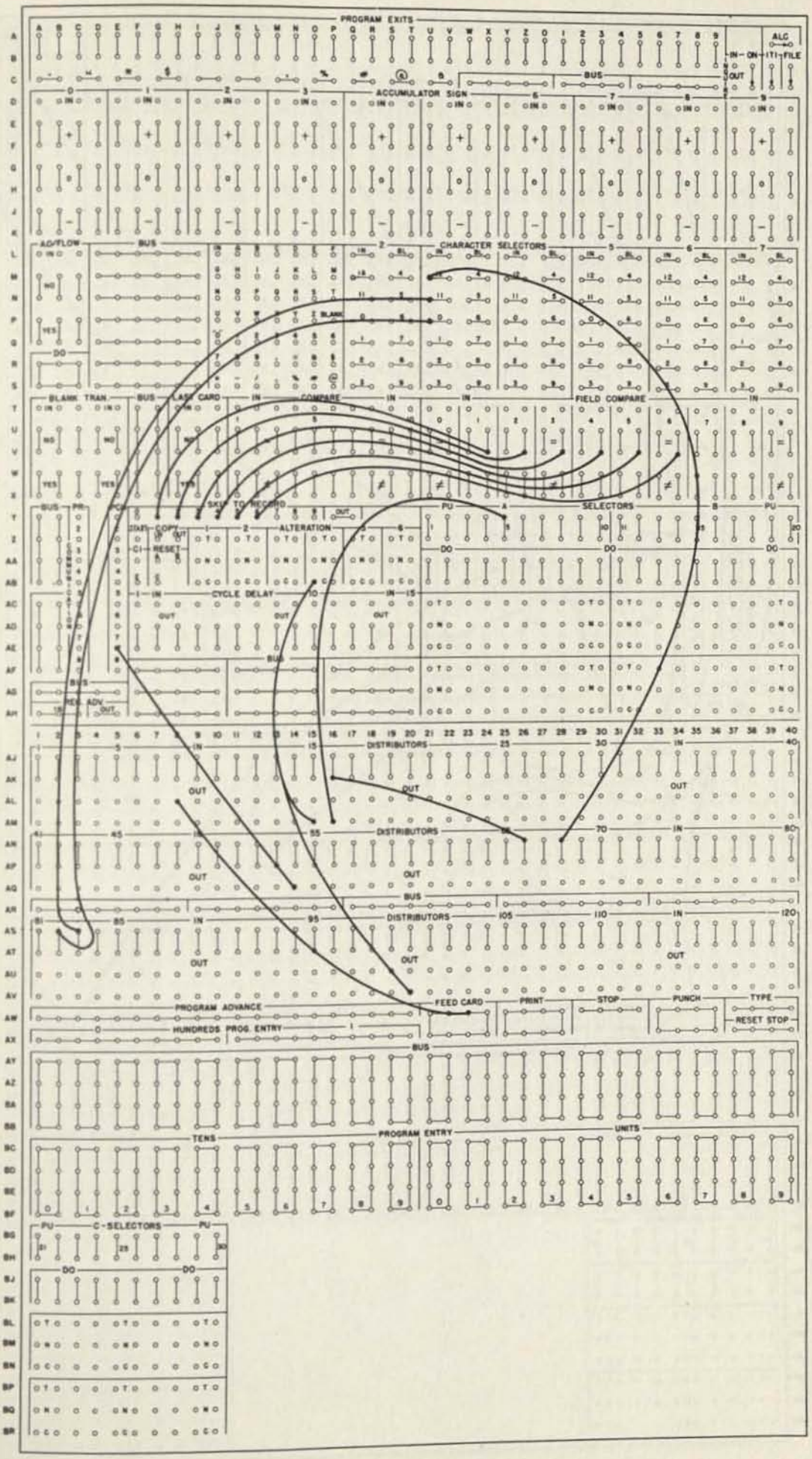

PROGRAM EXITS
ACCUMULATOR SIGN
BUS
AD/FLOW
CHARACTER SELECTORS
BLANK TRAN
LAST CARD
COMPARE
FIELD COMPARE
SKIP TO RECORD
CYCLE DELAY
ALTERATION
SELECTORS
DISTRIBUTORS
PROGRAM ADVANCE
FEED CARD
PRINT
STOP
PUNCH
TYPE
RESET STOP
HUNDREDS PROG. ENTRY
TENS
PROGRAM ENTRY
UNITS
C-SELECTORS

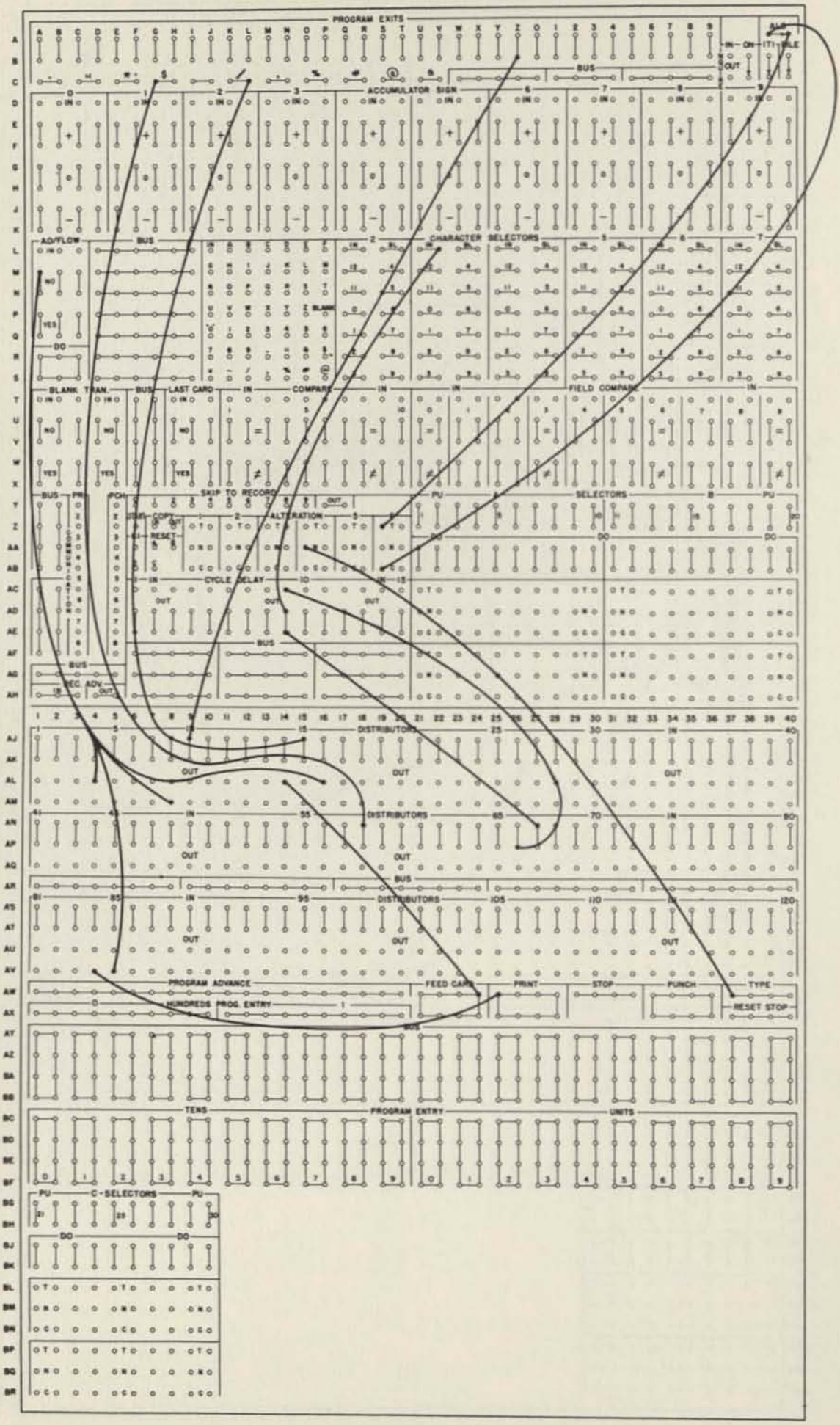

PROGRAM EXITS
ACCUMULATOR SIGN
CHARACTER SELECTORS
AD/FLO
BUS
BLANK TRAN.
LAST CARD
COMPARE
FIELD COMPARE
SKIP TO RECORD
ALTERATION
CYCLE DELAY
SELECTORS
DISTRIBUTORS
PROGRAM ADVANCE
HUNDREDS PROG. ENTRY
FEED CARD
PRINT
STOP
PUNCH
TYPE
RESET STOP
PROGRAM ENTRY
TENS
UNITS
C-SELECTORS

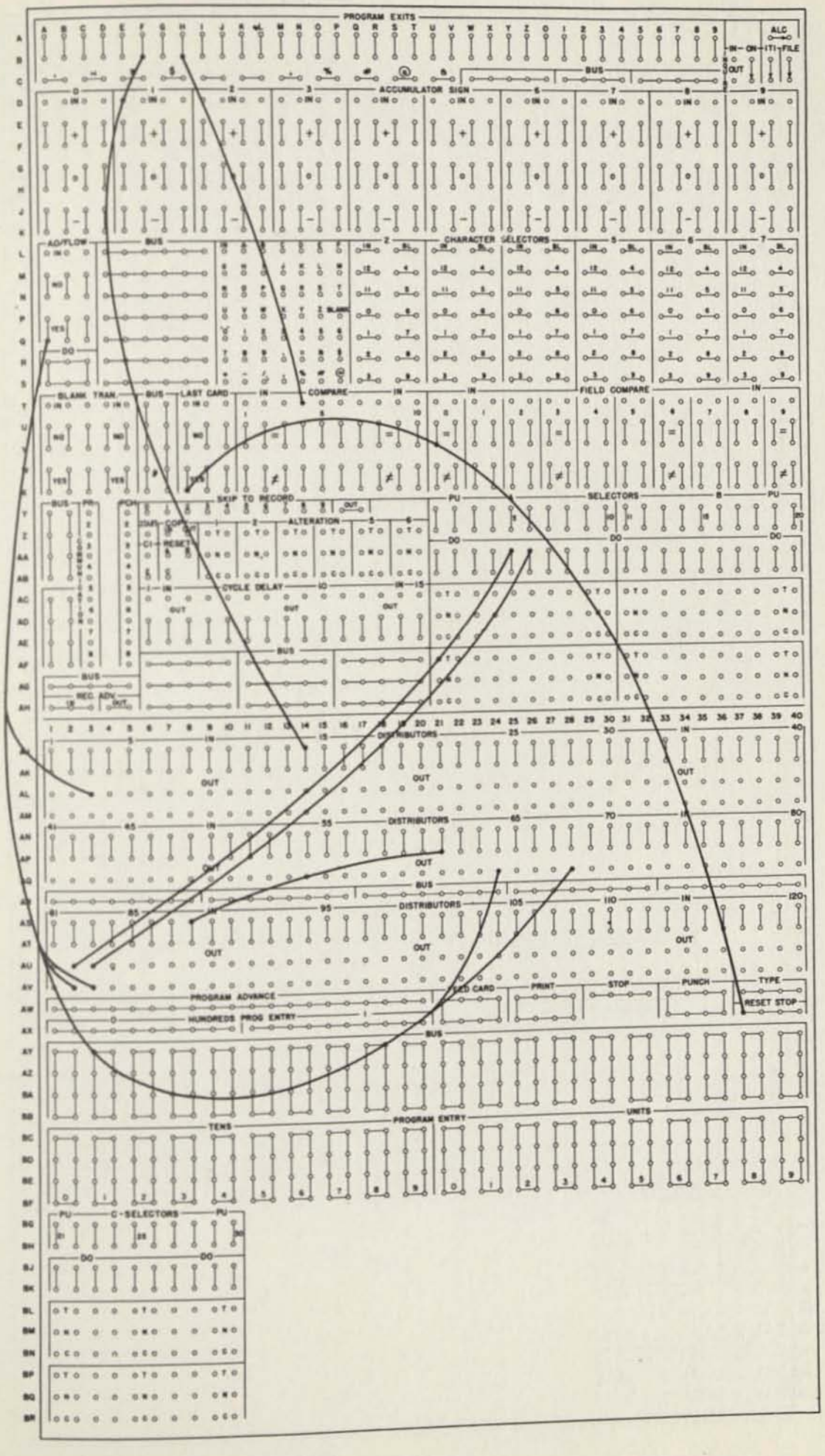

PROGRAM EXITS
ALC
IN
ON
IT1
FILE
OUT
BUS
ACCUMULATOR SIGN
AD/FLOW
NO
YES
DO
CHARACTER SELECTORS
BLANK TRAN.
LAST CARD
COMPARE
FIELD COMPARE
SKIP TO RECORD
CYCLE DELAY
ALTERATION
SELECTORS
PU
DC
REC. ADV.
DISTRIBUTORS
PROGRAM ADVANCE
FEED CARD
PRINT
STOP
PUNCH
TYPE
RESET STOP
HUNDREDS PROG. ENTRY
TENS
PROGRAM ENTRY
UNITS
C-SELECTORS

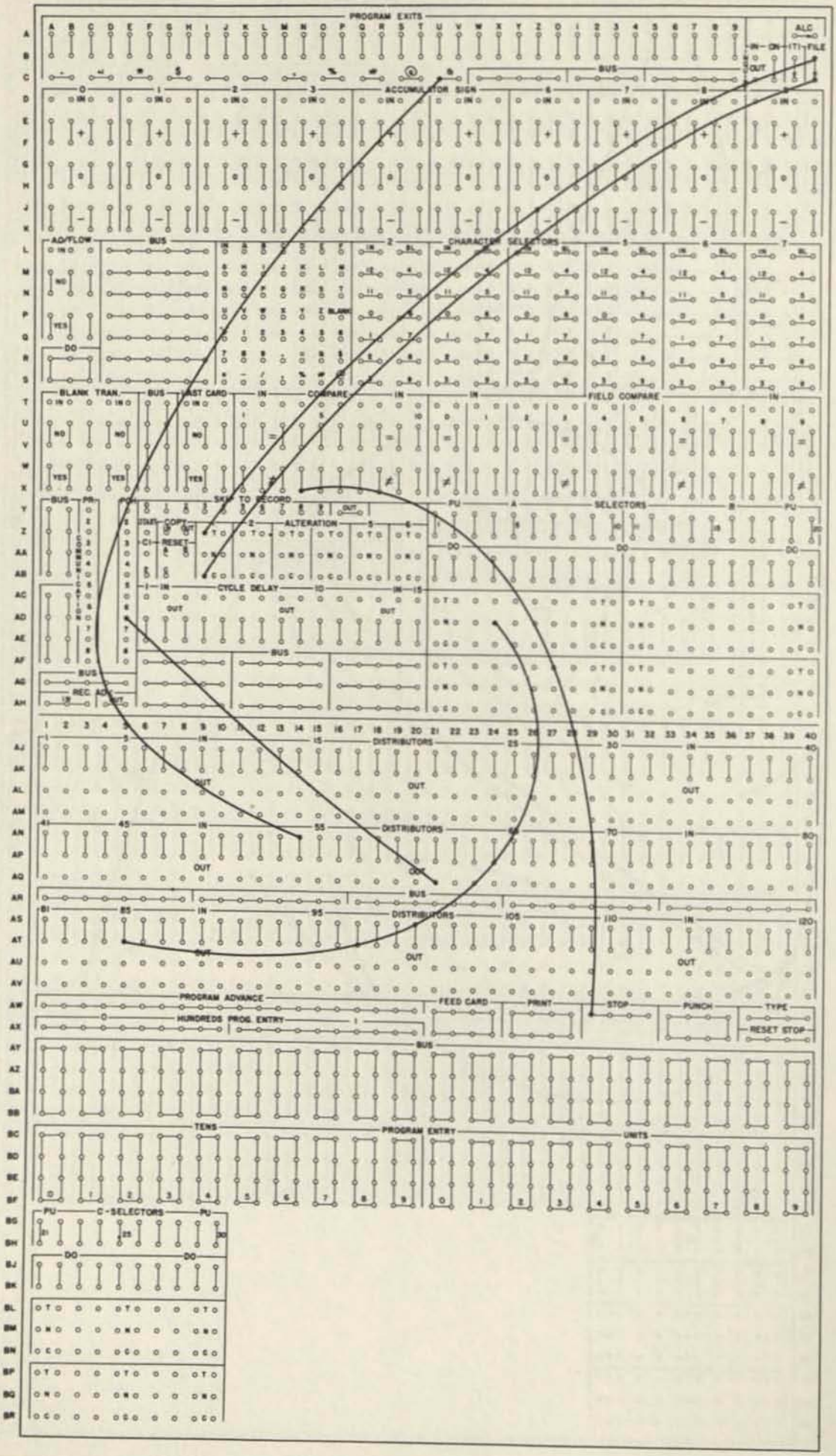

PROGRAM EXITS
ALC
(T)-FILE
BUS
ACCUMULATOR SIGN
ACTFLOW
CHARACTER SELECTORS
BLANK TRAN
LAST CARD
COMPARE
FIELD COMPARE
SKIP TO RECORD
ALTERATION
CYCLE DELAY
REC ADV
DISTRIBUTORS
PROGRAM ADVANCE
FEED CARD
PRINT
STOP
PUNCH
TYPE
RESET STOP
HUNDREDS PROG. ENTRY
TENS
PROGRAM ENTRY
UNITS
C-SELECTORS

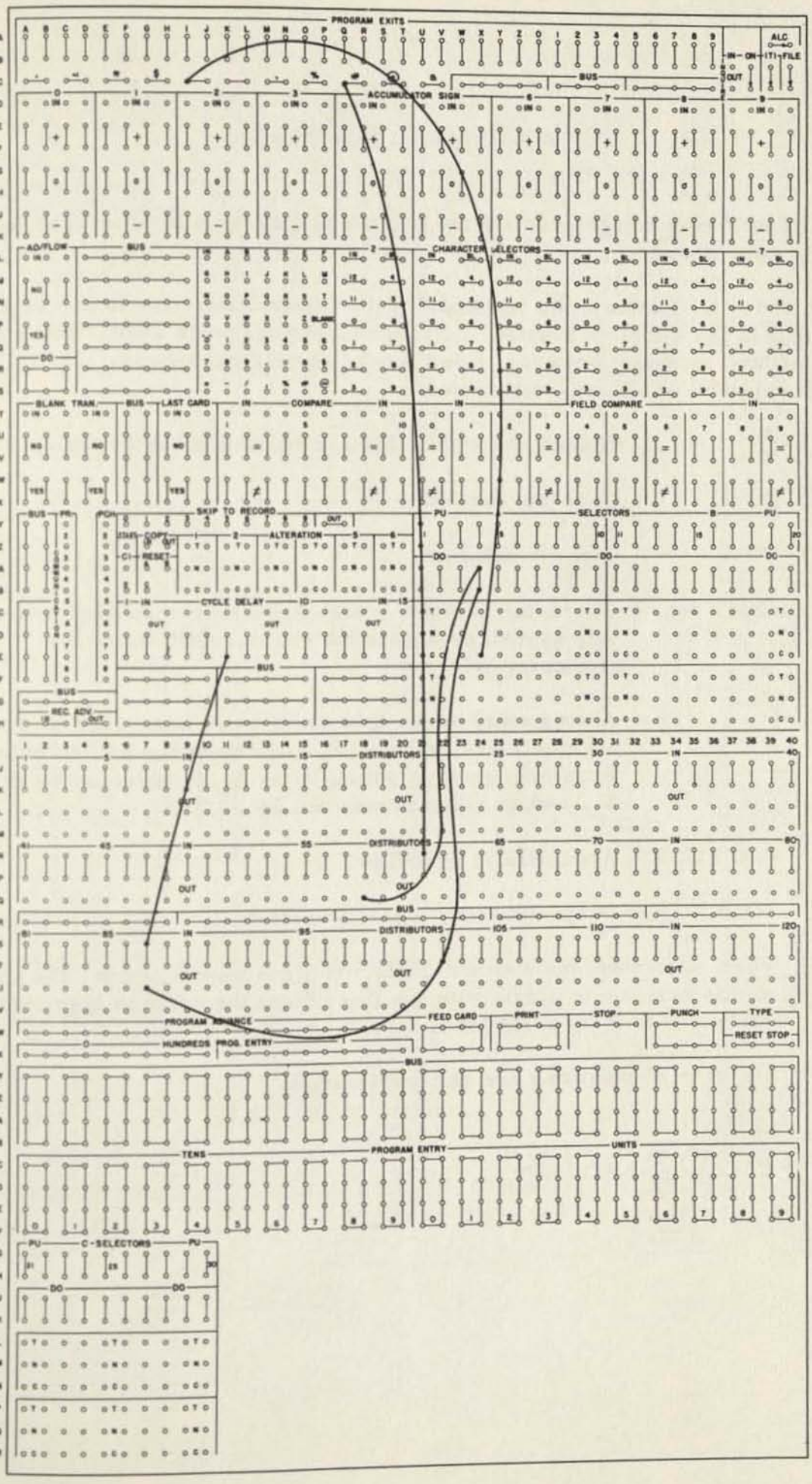

PROGRAM EXITS
ACCUMULATOR BRUN
CHARACTER SELECTORS
BLANK TRAN
LAST CARD
COMPARE
FIELD COMPARE
SKIP TO RECORD
CYCLE DELAY
ALTERATION
SELECTORS
DISTRIBUTORS
PROGRAM ADVANCE
FEED CARD
PRINT
STOP
PUNCH
TYPE
RESET STOP
HUNDREDS PROG. ENTRY
PROGRAM ENTRY
TENS
UNITS
C-SELECTORS


PROGRAM EXITS
ALG
FILE
ACCUMULATOR SIGN
CHARACTER SELECTORS
BLANK TRAN
LAST CARD
COMPARE
FIELD COMPARE
SELECTORS
SKIP TO RECORD
COPY
ALTERATION
RESET
CYCLE DELAY
DISTRIBUTORS
PROGRAM ADVANCE
FEED CARD
PRINT
STOP
PUNCH
TYPE
RESET STOP
HUNDREDS PROG. ENTRY
PROGRAM ENTRY
TENS
UNITS
C-SELECTORS


PROGRAM EXITS
ACCUMULATOR SIGN
CHARACTER SELECTORS
AD/FLOW
BUS
BLANK TRAN.
LAST CARD
COMPARE
FIELD COMPARE
SKIP TO RECORD
ALTERATION
CYCLE DELAY
RESET
SEL. ADV.
DISTRIBUTORS
PROGRAM ADVANCE
HUNDREDS PROG. ENTRY
FEED CARD
PRINT
STOP
PUNCH
TYPE
RESET STOP
TENS
PROGRAM ENTRY
UNITS
C-SELECTORS


PROGRAM EXITS
ACCUMULATOR SIGN
CHARACTER SELECTORS
BUS
ALC
ITI
FILE
AC/TFLOW
BLANK TRAN.
LAST CARD
COMPARE
FIELD COMPARE
SKIP TO RECORD
ALTERATION
CYCLE DELAY
REC. ADV.
SELECTORS
DISTRIBUTORS
PROGRAM ADVANCE
HUNDREDS PROG ENTRY
FED CARD
PRINT
STOP
PUNCH
TYPE
RESET STOP
TENS
PROGRAM ENTRY
UNITS
C-SELECTORS


PROGRAM EXITS
ALC
ACCUMULATOR SIGN
BUS
AD/FLOW
CHARACTER SELECTORS
BLANK TRAN.
LAST CARD
COMPARE
FIELD COMPARE
SKIP TO RECORD
ALTERATION
CYCLE DELAY
SELECTORS
DISTRIBUTORS
PROGRAM ADVANCE
FEED CARD
PRINT
STOP
PUNCH
TYPE
RESET STOP
HUNDREDS PROG. ENTRY
PROGRAM ENTRY
TENS
UNITS
C-SELECTORS
PU
DO


PROGRAM EXITS
ACCUMULATOR SIGN
CHARACTER SELECTORS
BUS
ALC
FILE
IN
ON
OUT
AD/FLOW
BLANK TRAN.
LAST CARD
COMPARE
FIELD COMPARE
SELECTORS
SKIP TO RECORD
COPY
RESET
ALTERATION
CYCLE DELAY
REC. ADV.
DISTRIBUTORS
PROGRAM ADVANCE
HUNDREDS PROG. ENTRY
FEED CARD
PRINT
STOP
PUNCH
TYPE
RESET STOP
PROGRAM ENTRY
TENS
UNITS
C-SELECTORS
PU
DO
NO
YES
BLANK

FUNCTIONAL DIAGRAMS

The following 6 pages contain the functional diagrams for the General Purpose Board.

In order to illustrate the functions of the General Purpose Board in the neatest and simplest fashion, certain conventions were used. Many are standard symbols; however, a few are of an original nature. Below is an appendix of all conventions used.

1) ◇ - Characters within this symbol designate Program Exits.

2) DO S_ - Drop-out Selector .

3) PU S_ - Pick-up Selector .

4) Char. Sel. - Character Selector - the characters encircled beneath this symbol designate the various OUT hubs in a character selector.

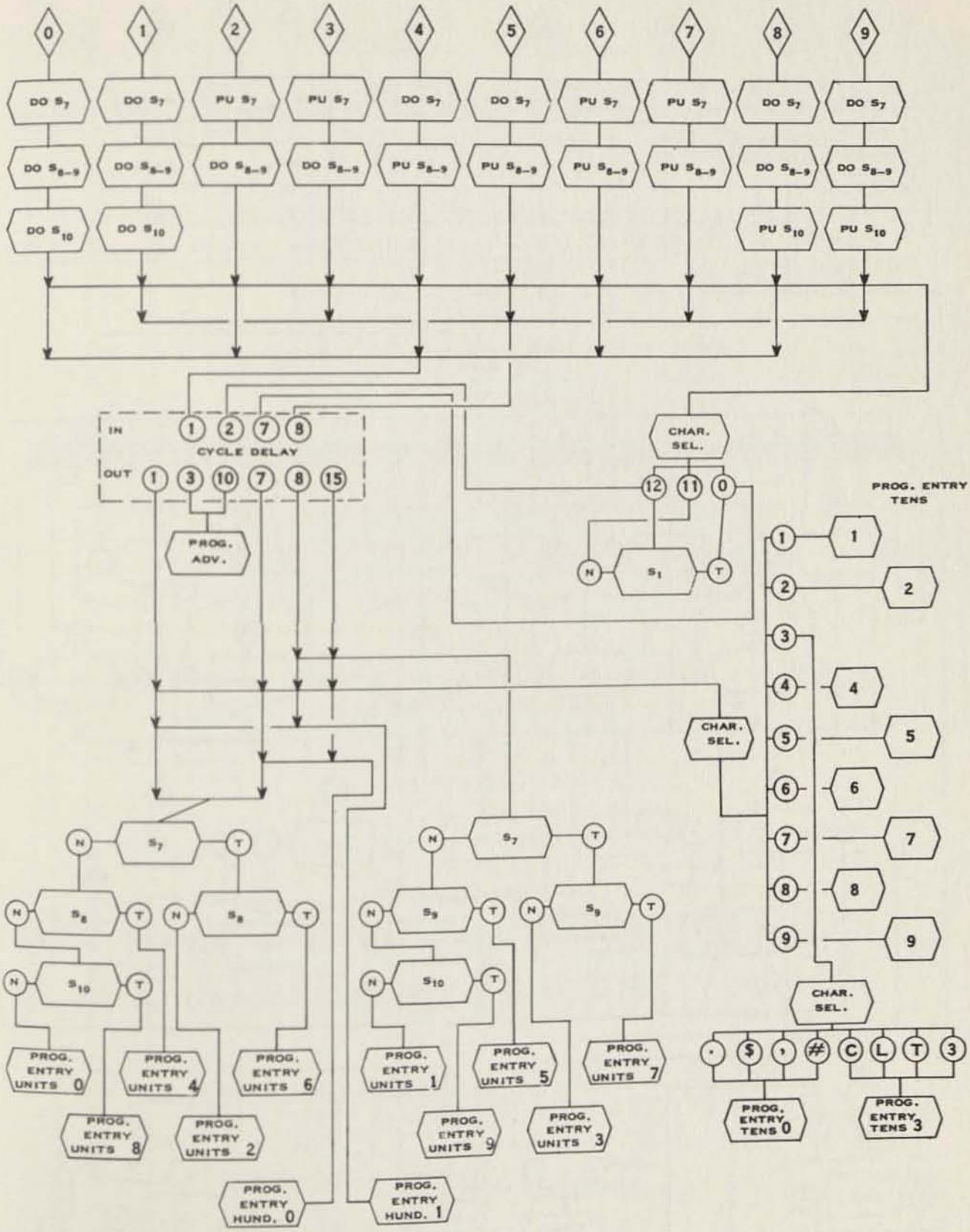
5) ⬭ - Among other uses, this symbol has been used to show accumulators, selectors, and alteration switch hubs. Accumulators are shown by an A_, a selector by an S_, and an alteration switch by ALT S_.

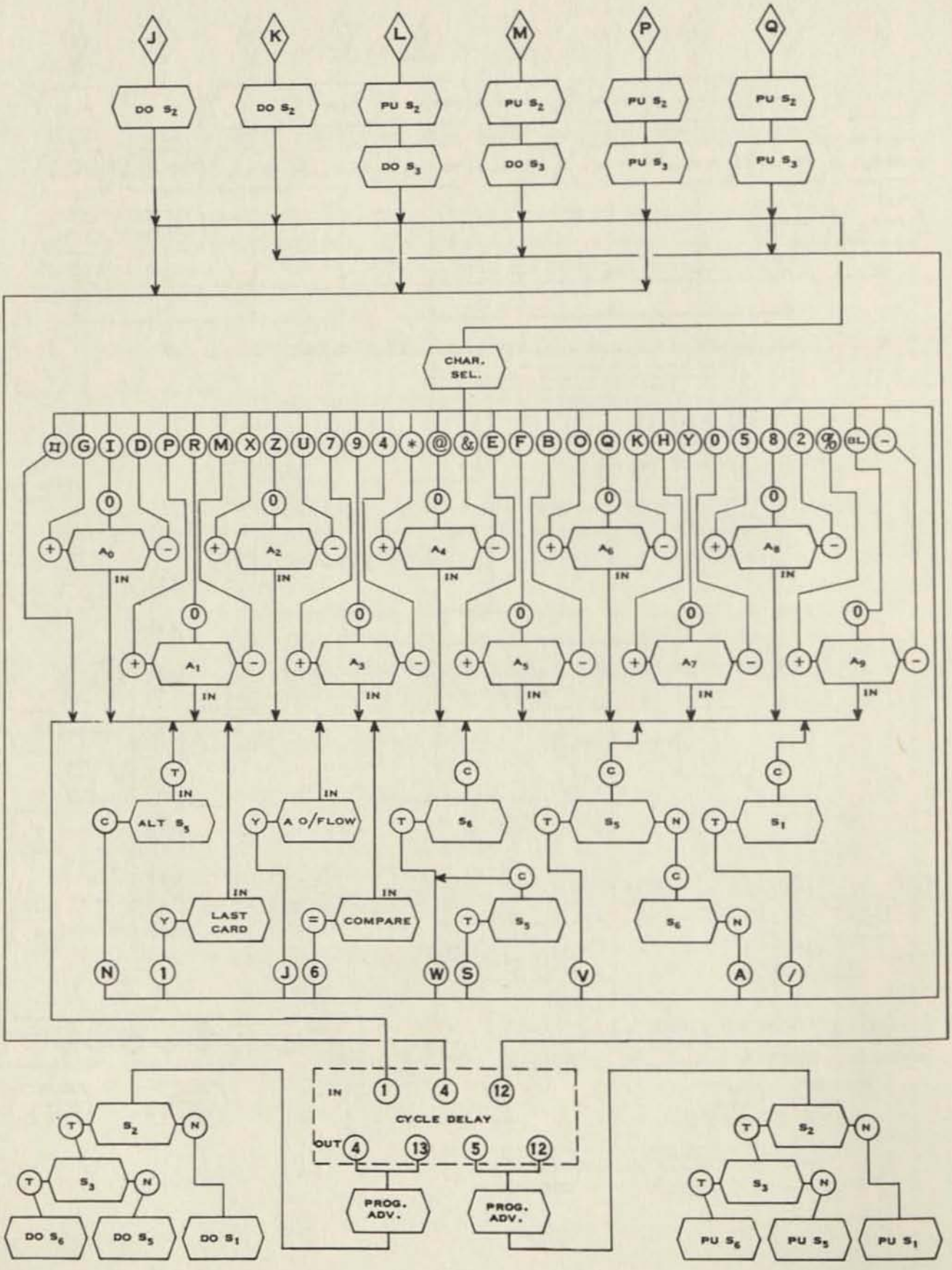
6) A/O FLOW - Accumulator Overflow.

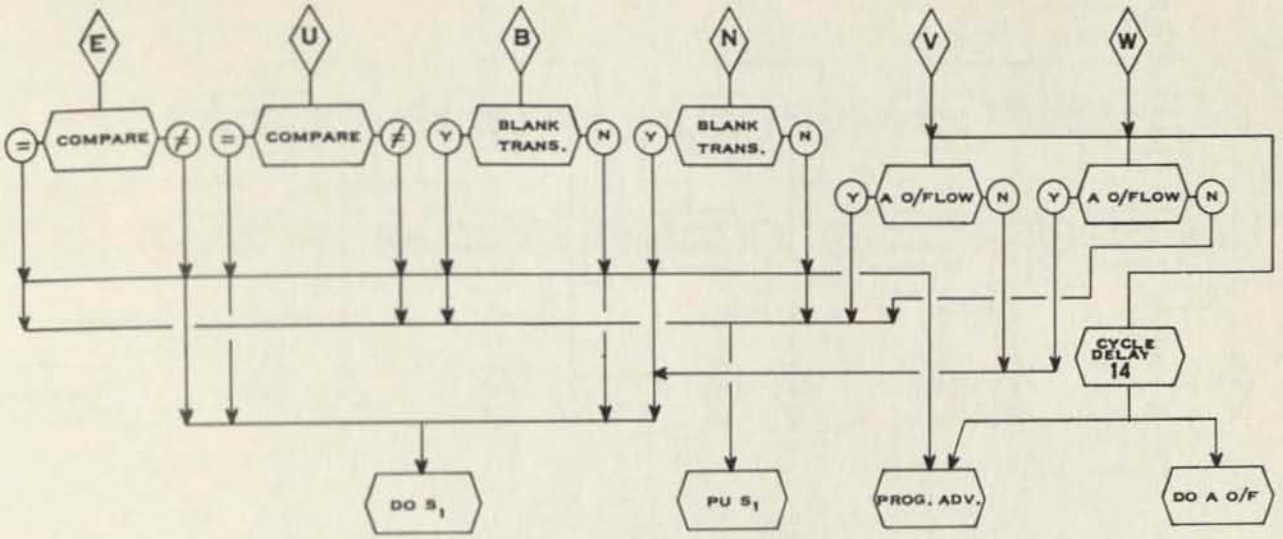
7) DO A O/F - Drop-out Accumulator Overflow.

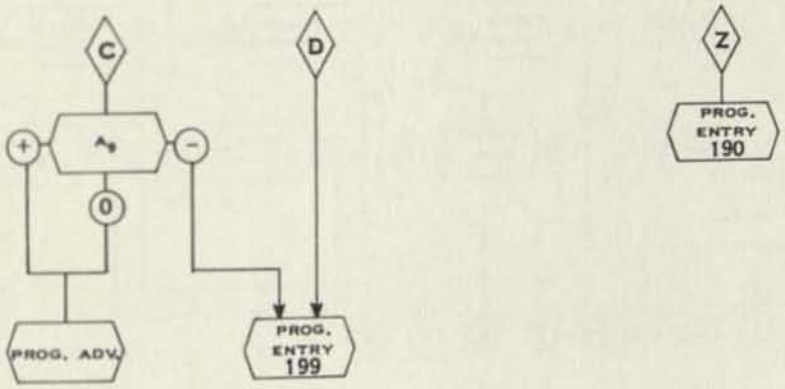
8) When more than one element performs a common function, the line to this function is single with the lines from the various elements connected to it by arrowheads.

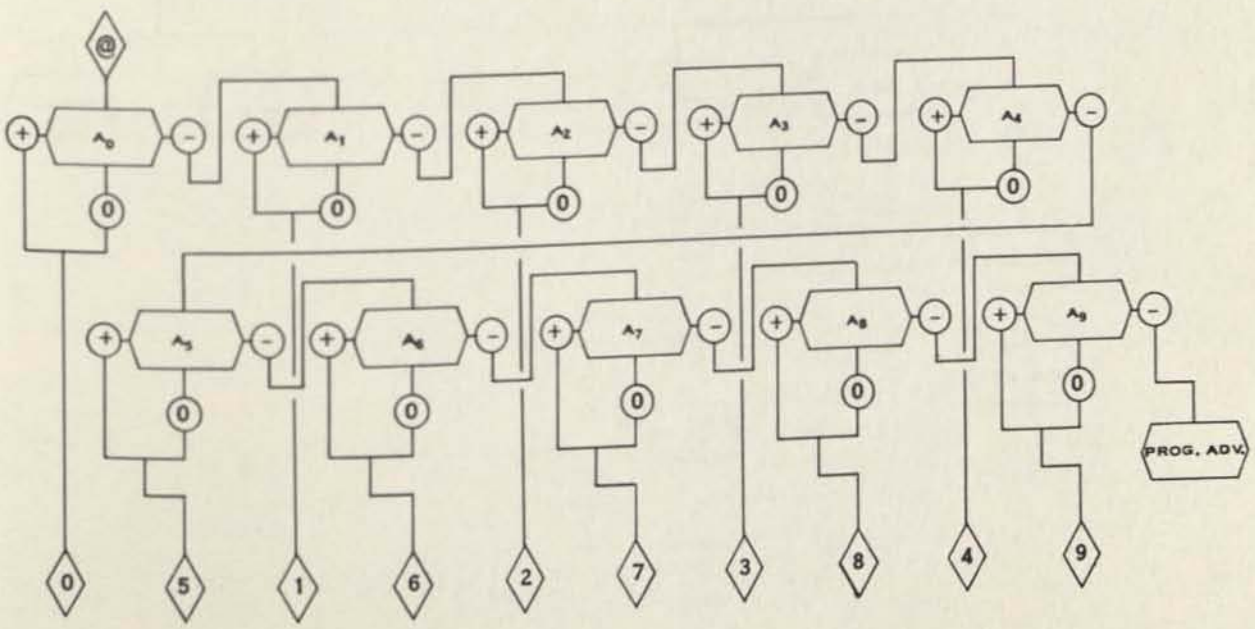
9) All other conventions are self-explanatory.

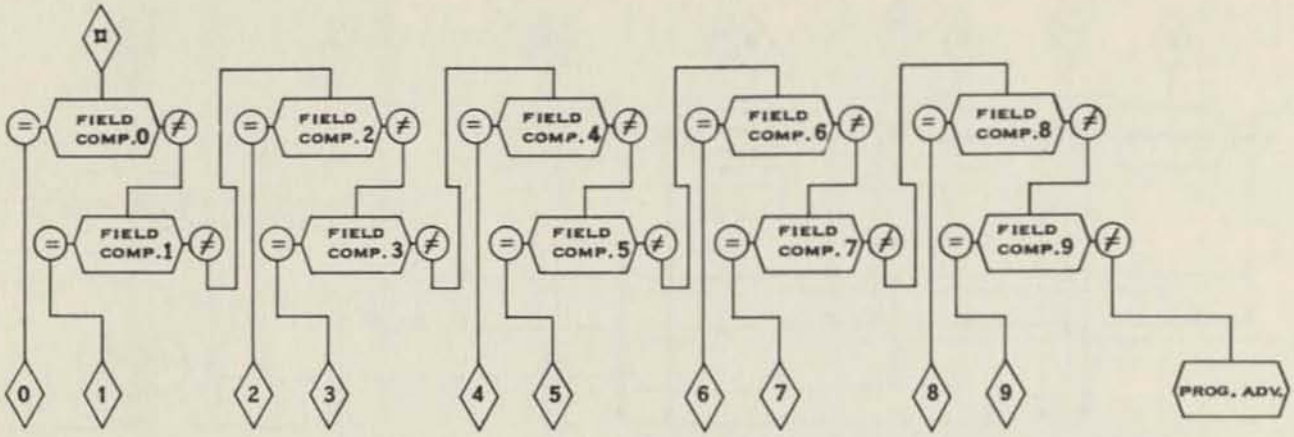

0
1
2
3
4
5
6
7
8
9
DO S7
PU S7
DO S8-9
PU S8-9
DO S10
PU S10
IN
1 2 7 8
CYCLE DELAY
OUT
1 3 10 7 8 15
PROG. ADV.
CHAR. SEL.
12 11 0
N
S1
T
PROG. ENTRY TENS
S7
S8
S9
S10
PROG. ENTRY UNITS 0
PROG. ENTRY UNITS 4
PROG. ENTRY UNITS 6
PROG. ENTRY UNITS 8
PROG. ENTRY UNITS 2
PROG. ENTRY UNITS 1
PROG. ENTRY UNITS 5
PROG. ENTRY UNITS 7
PROG. ENTRY UNITS 9
PROG. ENTRY UNITS 3
PROG. ENTRY HUND. 0
PROG. ENTRY HUND. 1
. $, # C L T 3
PROG. ENTRY TENS 0
PROG. ENTRY TENS 3

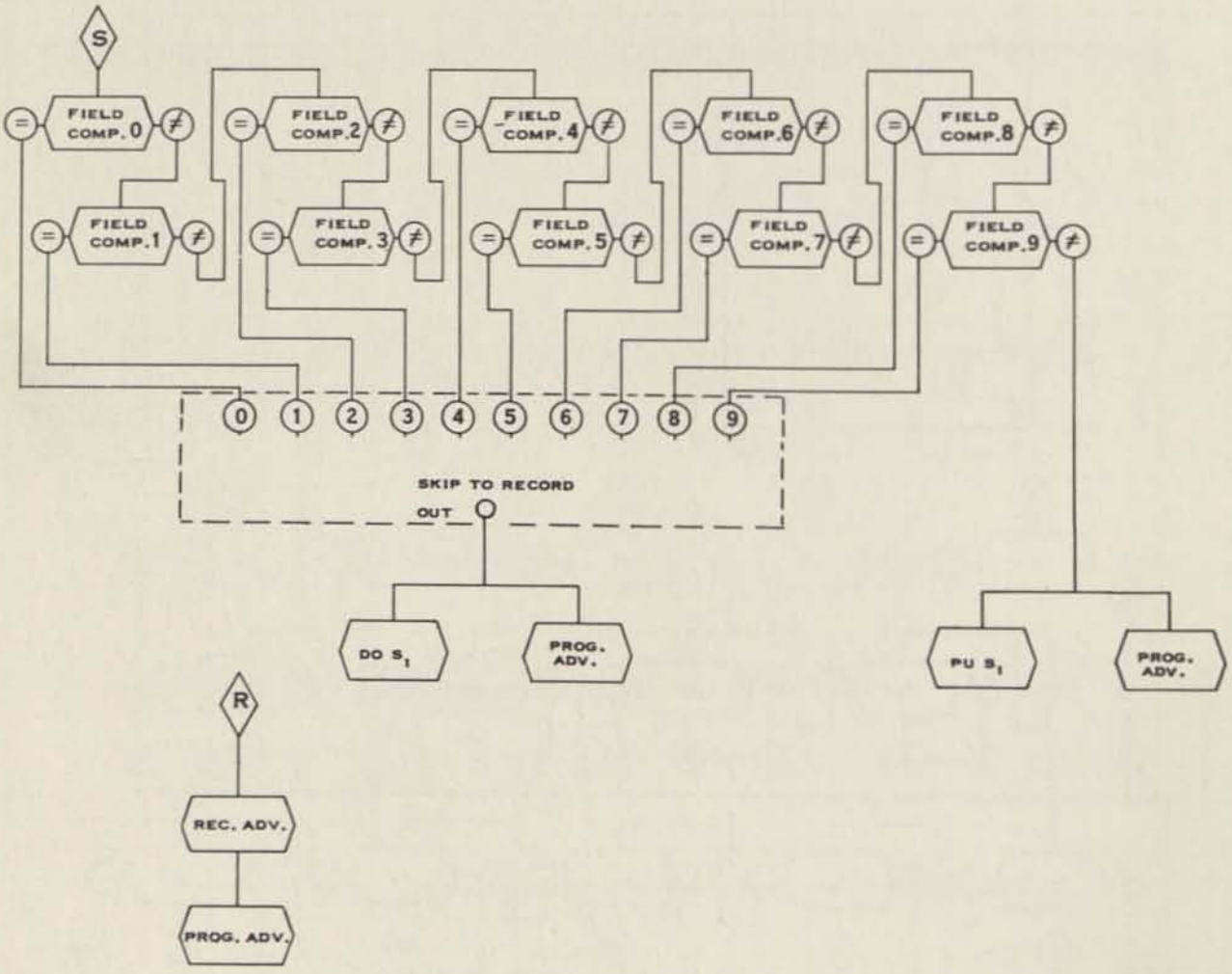

J
K
L
M
P
Q
DO S2
DO S2
PU S2
PU S2
PU S2
PU S2
DO S3
DO S3
PU S3
PU S3
CHAR. SEL.
A0
A1
A2
A3
A4
A5
A6
A7
A8
A9
IN
ALT S5
A O/FLOW
S6
S5
S1
LAST CARD
COMPARE
S5
S6
CYCLE DELAY
PROG. ADV.
PROG. ADV.
DO S6
DO S5
DO S1
PU S6
PU S5
PU S1

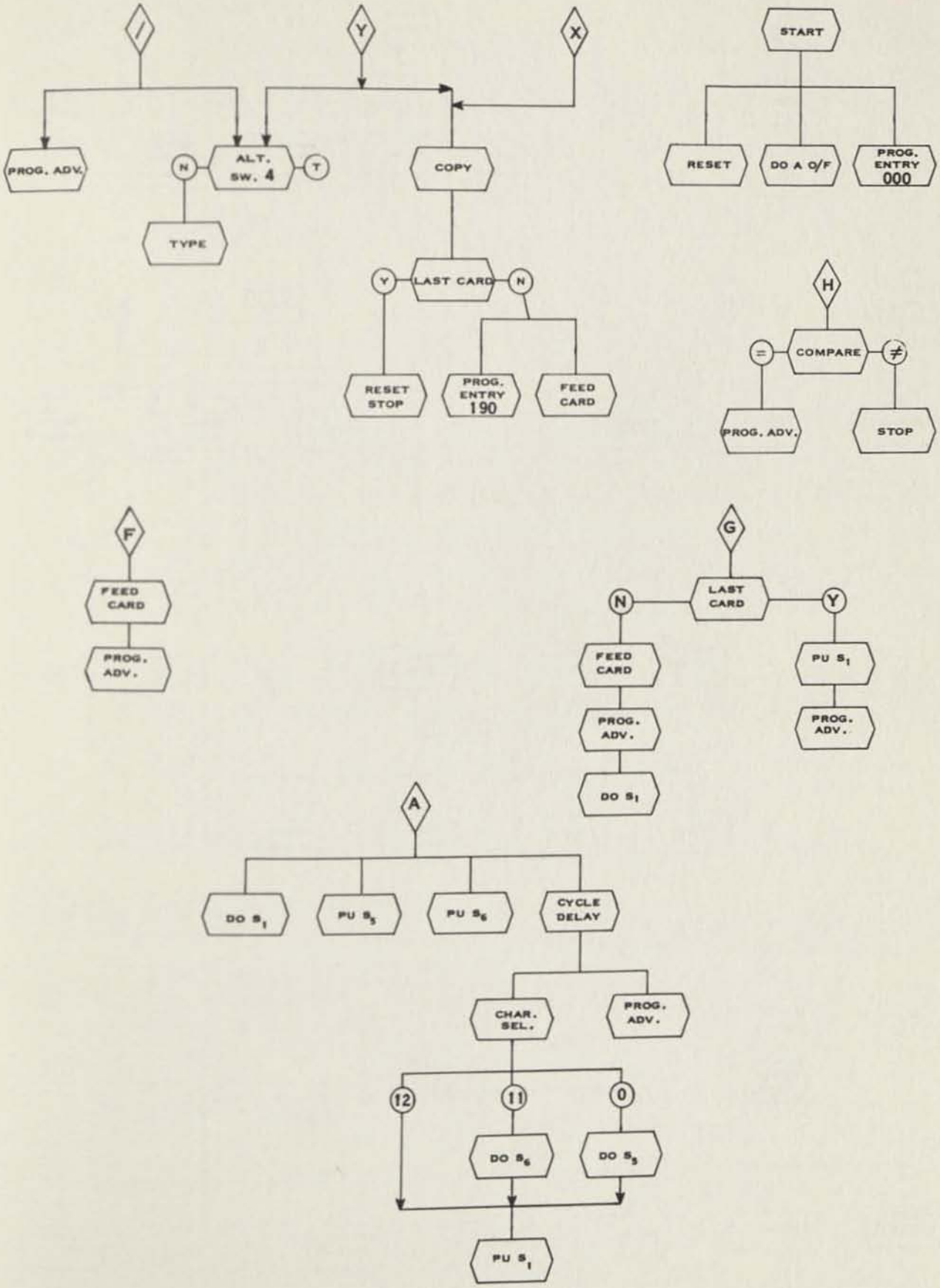

E
U
B
N
V
W
COMPARE
COMPARE
BLANK TRANS.
BLANK TRANS.
A O/FLOW
A O/FLOW
CYCLE DELAY 14
DO S1
PU S1
PROG. ADV.
DO A O/F


C
D
Z
PROG. ADV.
PROG. ENTRY 199
PROG. ENTRY 190

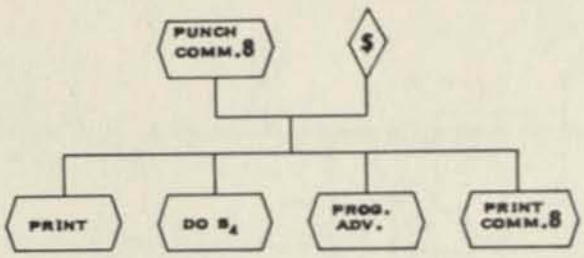

@
PROG. ADV.
0
5
1
6
2
7
3
8
4
9

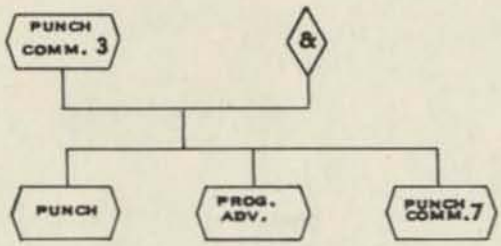

¤
FIELD COMP.0
FIELD COMP.1
FIELD COMP.2
FIELD COMP.3
FIELD COMP.4
FIELD COMP.5
FIELD COMP.6
FIELD COMP.7
FIELD COMP.8
FIELD COMP.9
0
1
2
3
4
5
6
7
8
9
PROG. ADV.

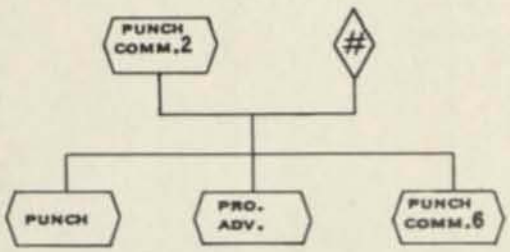

S
FIELD COMP. 0
FIELD COMP. 1
FIELD COMP.2
FIELD COMP. 3
FIELD COMP. 4
FIELD COMP. 5
FIELD COMP.6
FIELD COMP. 7
FIELD COMP.8
FIELD COMP.9
0 1 2 3 4 5 6 7 8 9
SKIP TO RECORD OUT
DO S1
PROG. ADV.
PU S1
PROG. ADV.
R
REC. ADV.
PROG. ADV.

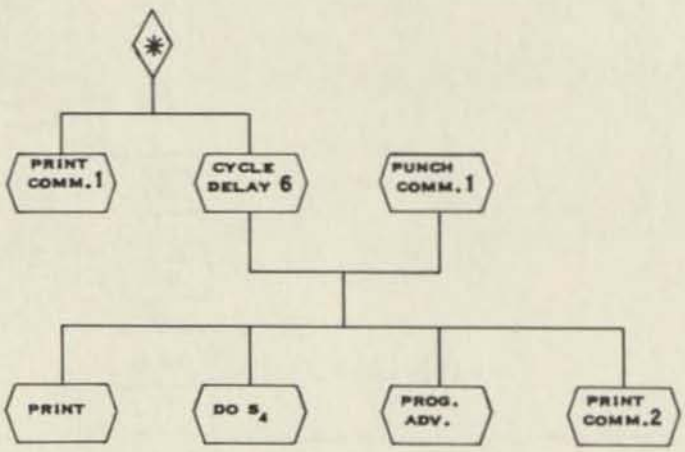

/
Y
X
PROG. ADV.
N
ALT. SW. 4
T
TYPE
COPY
Y
LAST CARD
N
RESET STOP
PROG. ENTRY 190
FEED CARD
START
RESET
DO A O/F
PROG. ENTRY 000
H
=
COMPARE
≠
PROG. ADV.
STOP
F
FEED CARD
PROG. ADV.
G
N
LAST CARD
Y
FEED CARD
PROG. ADV.
DO S1
PU S1
PROG. ADV.
A
DO S1
PU S5
PU S6
CYCLE DELAY
CHAR. SEL.
PROG. ADV.
12
11
0
DO S6
DO S5
PU S1

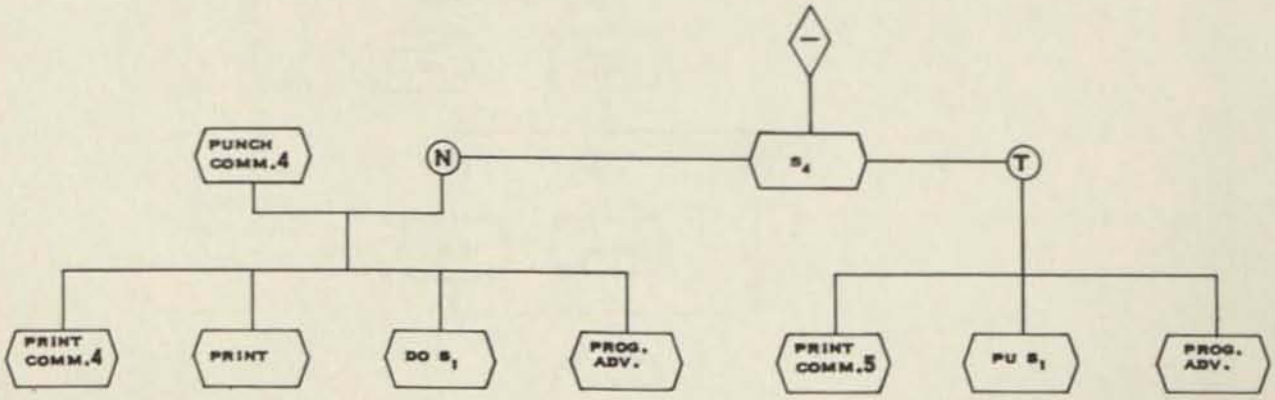
.
STOP
PROG. ADV.


PUNCH COMM.8
$
PRINT
DO S4
PROG. ADV.
PRINT COMM.8


PUNCH COMM. 3
&
PUNCH
PROG. ADV.
PUNCH COMM.7


PUNCH COMM.2
#
PUNCH
PRO. ADV.
PUNCH COMM.6


*
PRINT COMM.1
CYCLE DELAY 6
PUNCH COMM.1
PRINT
DO S4
PROG. ADV.
PRINT COMM.2


—
PUNCH COMM.4
N
S4
T
PRINT COMM.4
PRINT
DO S1
PROG. ADV.
PRINT COMM.5
PU S1
PROG. ADV.

IBM

International Business Machines Corporation
590 Madison Avenue New York 22 N Y

Printed in U.S.A. F28-2000